

LEVEL 12

TECHNICAL REPORT 80-3-97

# OVAL AND GENTREE: TWO APPROACHES TO PROBLEM STRUCTURING IN DECISION AIDS

DECISIONS AND DESIGNS INCORPORATED

Jonathan J. Weiss

April 1980

DTIC  
ECTE  
JUN 25 1980

This document has been approved  
for public release and sale; its  
distribution is unlimited.

ADA 086059

DDC FILE COPY.



## Command & Control Decision & Forecasting Systems Program

DEFENSE SCIENCES OFFICE: CYBERNETICS TECHNOLOGY  
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY  
Office of Naval Research • Engineering Psychology Programs

80 6 23 043

12

TECHNICAL REPORT 80-3-97

**QVAL AND GENTREE:  
TWO APPROACHES TO  
PROBLEM STRUCTURING IN DECISION AIDS**

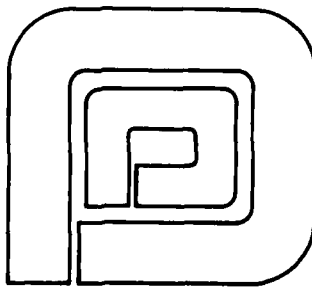
by

*Jonathan J. Weiss*

Prepared for

Defense Advanced Research Projects Agency  
Defense Sciences Office: Cybernetics Technology  
Contract N00014-79-C-0068

April 1980



**DECISIONS and DESIGNS, INC.**

Suite 600, 8400 Westpark Drive  
P.O. Box 907  
McLean, Virginia 22101  
(703) 821-2828

DTIC  
ELECTE  
JUN 25 1980

This document has been approved  
for public release and sale; its  
distribution is unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) TR-80-3-97	2. GOVT ACCESSION NO. AD-A086059	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) QVAL AND GENTREE: TWO APPROACHES TO PROBLEM STRUCTURING IN DECISION AIDS	5. TYPE OF REPORT & PERIOD COVERED (9) Technical Report	
7. AUTHOR(s) (10) Jonathan J. Weiss	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decisions and Designs, Inc. Suite 600, 8400 Westpark Dr., P.O. Box 907 McLean, VA 22101	8. CONTRACT OR GRANT NUMBER(s) (15) N00014-79-C-0068, ARPA Order-3668	
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency (CTO) 1400 Wilson Boulevard Arlington, VA 22209	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order No. 3668	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Engineering Psychology Programs 800 North Quincy Street Arlington, VA 22217	12. REPORT DATE April 1980 (11) Apr 80	
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited	13. NUMBER OF PAGES 149	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Decision analysis      Problem solving Decision theory      Computer applications Decision aids Decision making		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  Decision analysis is a technology which offers the decision maker a logical structure for solving a specific problem. Through the process of decision analysis, the decision maker may select from several options the one that best suits his/her needs. In practical applications, most decision analytic techniques require a specially trained decision analyst and a sizeable amount of time and expense. This report describes two computer-assisted decision aids which are designed to perform preliminary decision analyses in the -> next page		

DD FORM 1473  
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

cont absence of a professional decision analyst and for a restricted set of decision problems.

The first decision aid, QVAL (Quick Evaluation), has been implemented on an IBM 5110 portable computer and is ready for operation on an experimental basis. QVAL is an interactive program designed to probe the user's memory for a complete set of attributes together with the scores and weights for a consistent evaluation of a set of options.

The second decision aid, GenTree (Generic Tree Structuring), is still in the planning stage, but promises to be very valuable in storing experience and knowledge gained while working on problems and applying it to guide the analysis of subsequent ones. By utilizing prior knowledge stored in the GenTree database, this decision aid would approximate the performance of a decision analyst on familiar problem types. In contrast, QVAL would more resemble the performance of the decision analyst on a completely new application area.

With the continuing progress being made in the fields of artificial intelligence and interactive computer graphics, more sophisticated computer-assisted decision aids of this nature may be designed. However, QVAL and GenTree mark important first steps in bringing the computer's speed, accuracy, patience, and memory together with the user's knowledge, values, and judgment, to improve the timeliness and the quality of important decisions.

Accession For	
NTIS - QVAL	<input checked="checked" type="checkbox"/>
DDC TAB	
Exempted Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



## CONTENTS

	<u>Page</u>
1.0 EXECUTIVE SUMMARY	1
2.0 BACKGROUND	4
2.1 Motivation	4
2.2 Detailed Definition of the Problem	5
2.3 Uses, Goals, and Constraints for QVAL and GenTree	6
3.0 OVAL: A QUICK INTERACTIVE EVALUATION PROGRAM	8
3.1 Overview	8
3.2 Specification of Options	8
3.3 Identification of Utility Attributes	10
3.4 Preliminary Elicitation of Scores	15
3.5 Assignment of Attribute Weights	19
3.6 Calculations and Summary Display	26
3.7 Iterative Expansion of the Model	29
3.7.1 Additional attributes affecting the best option	29
3.7.2 Attributes affecting selection of "next-best" options	32
3.8 Hierarchical Organization of Utility Attributes	36
3.9 Sensitivity Analysis and Editing	44
4.0 A USER'S GUIDE TO DDI'S CURRENT VERSION OF QVAL	48
4.1 Loading and Initial Routine	48
4.2 Load Model Routine	49
4.3 New Model Routine	49
4.3.1 Initial elicitation of attribute names	51
4.3.2 Assessment of initial scores	51
4.3.3 Elicitation of relative attribute weights	55
4.4 Main Menu Routine	58
4.5 EXIT Routine	61
4.6 DISPLAY Routine	62
4.7 MODIFY Routine	63

# CONTENTS (Cont'd)

	<u>Page</u>
4.7.1 MODIFY WEIGHTS	64
4.7.2 MODIFY SCORES	64
4.7.3 DISPLAY CURRENT VALUES	69
4.7.4 ADD AN OPTION	69
4.7.5 DELETE AN OPTION	69
4.7.6 DELETE AN ATTRIBUTE	71
4.7.7 MODIFY OPTION LABELS	72
4.7.8 MODIFY ATTRIBUTE LABELS	74
4.8 SENSITIVITY ANALYSIS Routine	74
4.9 ADD MORE DIMENSIONS and NEXT-BEST OPTION Routines	76
4.9.1 ADD MORE DIMENSIONS	76
4.9.2 NEXT-BEST OPTION	78
5.0 SUMMARY OF THEORETICAL AND METHODOLOGICAL CONSIDERATIONS FOR FUTURE QVAL DEVELOPMENT	82
5.1 Helping the User Generate a Complete, Well-Structured Set of Attributes	82
5.2 Alternatives to Relative Scoring Methods	83
5.3 Reducing Biases in Elicitation	87
5.4 Top-Down Versus Bottom-Up Weighting in Hierarchical Utility Models	89
5.5 Incorporating Non-Additive Interactions into the Models	92
5.6 More Sophisticated Clustering Routines	96
5.7 Explanation, Interpretation, and Display of Results	96
5.8 Artificial Intelligence Programs to Aid Decision Modeling	98
5.9 Accommodating Imprecision in Assessments	99
6.0 GENTREE: A CONTENT-ORIENTED AID TO DECISION PROBLEM STRUCTURING	103
6.1 Introduction	103
6.2 Theoretical Analysis	107
6.2.1 Data representation system	109
6.2.2 Semantic context framework	112
6.2.3 Overall mode of operation	114
6.2.4 Alternatives and extensions	117

CONTENTS  
(Cont'd)

	<u>Page</u>
6.3 A Notional Example	118
6.4 GenTree Summary	129
7.0 CONCLUSION	131
REFERENCES	134
APPENDIX A. IBM 5110 EQUIPMENT AND OPERATION	A-1
DISTRIBUTION LIST	



## 1.0 EXECUTIVE SUMMARY

Decision analysis is a valuable technology for helping a decision maker select which of several options, overall, best satisfies his various goals and requirements. Unfortunately, the techniques of decision analysis are typically complex enough to require three items: the assistance of a specially trained decision analyst, a considerable amount of time, and the funds necessary to perform a thorough analysis. These restrictions often prevent or discourage the use of decision analysis where it would otherwise be highly beneficial.

This report describes two different ways of using computer-implemented routines to perform preliminary decision analyses (or simple, complete ones) in the absence of specially trained personnel, for a restricted category of problems: the evaluation of multi-attributed alternatives without explicit use of probabilities to model uncertain events. While no claim is made that either of these computer methods could completely replace the trained analyst, it is likely they can provide the decision maker with a valuable initial focus, while reducing the actual amount of analyst time spent on a problem; and in the event of limited time, resources, or analyst availability, the preliminary analysis alone would provide useful insights which could improve significantly upon otherwise unaided decisions.

The first of the two methods described here is a program called QVAL (Quick Evaluation), which has been implemented on an IBM 5110 portable computer and is ready for operation on an experimental basis. QVAL is a user-oriented, interactive program which elicits from the decision maker the list of options available, the attributes which impact on overall value, the scores for each option on each of the attributes, and relative importance weights for the attri-

butes. Based on the elicited information, QVAL can display a component analysis of the overall values, perform sensitivity analysis, refine the existing model, and interact with the user to produce a hierarchical structure which organizes the list of attributes into a more meaningful and useful format. Section 2.0 provides a theoretical overview; Section 3.0 is a theoretical and operational analysis of QVAL; Section 4.0 is a functional User's Guide to the computer program version of QVAL currently in operation at Decisions and Designs, Incorporated (DDI); and Section 5.0 is a summary of the most promising directions for future development of the QVAL concept.

A second concept for computer-aided decision problem structuring is GenTree (Generic Tree Structuring), which is still in the planning stage, but promises, if successfully implemented, to be extremely valuable in storing experience and knowledge gained while working on problems, and applying it to guide the analysis of subsequent ones. The special feature which characterizes the GenTree approach is a database containing problem-specific information of two types: a semantic system which allows the current problem to be categorized in relation to previously analyzed ones; and a "knowledge" system which organizes the content of GenTree's analyses into a logical framework (such as the case grammar described by Fillmore, 1968) for access in subsequent problems. By utilizing this prior knowledge, GenTree would provide a better approximation to the performance of a decision analyst on familiar problem types (while QVAL would more resemble the performance of the decision analyst on a completely new application area). Section 6.0 describes the theory behind the GenTree concept, and presents a sample of how a GenTree system might operate.

The two decision aids described here, QVAL and GenTree, represent an initial effort at computer-aided decision

analysis beyond the simple clerical and computational functions of current aids. Eventually, by incorporating the continuing progress in artificial intelligence and in interactive computer graphics, a far more sophisticated system can be foreseen, approaching or even surpassing the capabilities of today's decision analysts; but at present, QVAL and GenTree mark important first steps in bringing the computer's speed, accuracy, patience, and memory together with the user's knowledge, values, and judgment, to improve the timeliness and the quality of important decisions.



## 2.0 BACKGROUND

### 2.1 Motivation

One of the most frequently applied genres of decision analysis involves multi-attribute evaluation of utilities without explicit modeling of uncertain events as probabilities. Using general-purpose software such as DDI's EVAL or HIVAL programs, the decision analyst can minimize the mechanical aspects of calculation and "housekeeping" (keying in data, editing, storing, retrieving, and displaying results); however, the analyst receives no assistance in the technical processes of eliciting the model and making the needed assessments--the computer's role is limited to that of an efficient, rapid, and infinitely patient clerk.

One can visualize a number of situations in which it would be desirable for the computer to take a more active role in assisting or relieving the decision analyst, whenever a portion of the analysis can be appropriately programmed. At a minimal level, such a capability would greatly improve the speed and the smoothness of the analysis, leaving the analyst free to concentrate on the more important technical issues. A slightly more sophisticated version could take on the role of a second analyst in the room, performing the routine portion of the modeling with only occasional intervention from the trained decision analyst (who could then allocate his time to interacting with the clients or dealing with exceptional problems). Finally, a fully developed program could be used to work directly with the client in the absence of any decision analyst, performing a complete--but tentative--analysis whose results could be applied immediately, or used as a starting point for a subsequent, more detailed, session with the decision analyst.

The effectiveness of a quick interactive evaluation routine will depend on a number of factors: the program's degree of sophistication, the user's ability to respond in an appropriate manner to the questions asked, the complexity and tractability of the particular problem, and the way in which results are interpreted and applied. In certain cases, factors such as time, cost, or analyst availability might preclude any other more extensive analysis; in others, the program could be used as a "first cut" at the problem, or as an independent (hopefully redundant) modeling effort to enhance the validity of the conventional analysis. In any event, a quick, painless evaluation routine should represent a low-cost, low-risk option with potentially great benefits.

## 2.2 Detailed Definition of the Problem

In the context of this report, we shall assume that a small amount of informal discussion has already taken place, enough to ensure that a multi-attribute (possibly hierarchical) utility evaluation model is likely to prove appropriate to the given problem. Furthermore, we shall assume that the set of realistic options can be reduced to a small number (up to about eight) of discrete alternatives, each sufficiently familiar that it can be evaluated meaningfully.

Under the program's guidance, the user (decision maker) should be able to generate a reasonably exhaustive set of evaluation criteria, assign scores to each of the options, and assess importance weights to the criteria. These values should be automatically displayed, along with their implications, in terms of overall utility scores. Continuing the analysis, the computer should then guide the user through any indicated sensitivity analysis, editing, expansion of the model, and other procedures for improving the model's correctness, completeness, or clarity. Finally, if the

number of criteria is large, the program should guide the user in organizing the attributes hierarchically, so that meaningful groups of attributes can be readily displayed and studied.

### 2.3 Uses, Goals, and Constraints for QVAL and GenTree

As presently conceived, QVAL and GenTree will have two major types of application: (1) they will serve as a self-contained decision aid for situations when time, cost, or availability constraints preclude a more complete analysis with trained decision analysts present; and (2) they will be useful as a preliminary structuring device, as a "front end" for a more complete analysis. Other potential applications might include training newcomers in decision-analytic procedures, or running independent, parallel analyses of a single problem with different subjects.

The goals which would contribute to the above applications would represent new developments (or new applications of old developments) in three areas: the elicitation of model components (attribute names, scores, and weights); the organization of those components into a meaningful and useful structure; and the effective display, refinement, and communication of the final results. QVAL and GenTree will be judged a success to the degree that they perform these functions rapidly, effectively, and painlessly, with a minimum of prior training on the part of the user.

The anticipated context for the application of these aids assumes a few constraints which should be mentioned here. First, the user should not be required to have any technical decision-analytic training, nor should it be necessary to refer to a lengthy user's guide or training manual: a brief introduction should suffice. Any further definitions or instructions should be accessible directly as part of the program.



Furthermore, the user should not be required to perform any off-line pencil-and-paper tasks or calculations, prior to or during the analytic process. This precludes, for example, the extensive off-line modeling that must precede the use of general-purpose programs such as EVAL (Allardyce et al., 1979).

One further constraint involves the availability and use of substantive knowledge gained through previous analytic studies: QVAL will not require such knowledge, and will not be designed in such a way as to incorporate such knowledge even if it were available. In other words, QVAL's procedure is deliberately constrained to be content-free (and, therefore, equally applicable to novel or familiar problems). GenTree, on the other hand, explicitly encodes, stores, and utilizes substantive information while interacting with the user to produce a valid hierarchically structured utility model.

### 3.0 QVAL: A QUICK INTERACTIVE EVALUATION PROGRAM

#### 3.1 Overview

This section describes the components of the QVAL process in detail, referring as much as possible to the operational and theoretical aspects of the procedures, rather than to their specific implementation on the computer. Section 4.0 provides a systematic description of a version of QVAL which has been implemented on an IBM 5110 computer for illustration and evaluation; the emphasis of the current section, however, is on the general foundations behind QVAL's approach.

Each of the following subsections will address one of QVAL's component tasks, including the following information:

- o a definition of the task component being described;
- o the method currently adopted by QVAL to perform that task;
- o a theoretical rationale, explaining the reasons for performing the task as stated, and indicating other possible approaches whenever it would be useful; and
- o any recommendations for future improvements or new developments which might enhance performance or expand QVAL's capabilities.

#### 3.2 Specification of Options

Task description. The first step in an evaluation is to develop a list of the options being compared, and to

specify whether that list is completely exhaustive, or merely a subset of the total number of possible choices. Since we have assumed a small number of discrete options, a simple listing of the choices will suffice; if, however, there were many options, or if the options had to be selected from a continuous space, option generation would become a serious problem, meriting an extensive research effort.

Current procedure. The user simply types in the names of the options being evaluated, one at a time.

Theoretical analysis. Because of its simplicity, the current procedure requires little discussion. However, it might be appropriate to mention a few more sophisticated approaches which might be used to reduce a larger set of options to the size QVAL was designed for.

One source which might generate large numbers of options is factorial design: each option corresponds to a combination of factors which can vary independently, so that the total space of options is equivalent to the set of all possible combinations of values on the various factors. So, for example, it might be possible to describe television sets in terms of screen size, color/black-and-white, portability, remote control features, etc. If all of the possible combinations are in fact available, perhaps a different analytical model (such as DDI's "DESIGN" model, as described in Gulick et al., 1979) would be more appropriate. But if there are several infeasible combinations (and yet too many feasible ones to permit simply listing them), it might be possible to use the feature descriptions as the basis for a hierarchical clustering analysis (see Hartigan, 1975), reducing the choice from a multitude of alternatives to a few generic clusters (with the option of subsequently deciding which specific item within the selected cluster was the best).

In the event that a suitable factorial description is not obvious, another possible approach might be to start the analysis with a few likely candidates, but after partial analysis, temporarily "shelving" the less promising options, and replacing those with new ones from the list.

It is clear that either of these more sophisticated means of reducing a large list of options could require a substantial investment in time and effort would be necessary to develop a sufficiently general procedure to be worth incorporating into a QVAL-type aid. Therefore, no attempt has been made to implement this capability under the current system.

Recommended future development. If the need justifies the development effort, and it appears feasible to incorporate a factorial analysis with hierarchical clustering into a future version of QVAL, such a facility could greatly augment the applicability of this interactive decision aid. In any event, it will probably prove worthwhile to incorporate some method for screening and selectively evaluating large numbers of alternatives.

### 3.3 Identification of Utility Attributes

Task description. Once a list of options has been specified, the next step in a multi-attribute utility analysis is to identify those differences among the options which might contribute significantly to a relative assessment of their net worth. Having identified such attributes, the user will subsequently proceed to define a rating scale for each attribute (i.e., a correspondence between the options' performance on each attribute and a set of numerical ratings); to assess rating scores for each option with respect to each attribute; and then to establish equivalences which will allow him to compare and combine scores on the various attributes.

In order to contribute meaningfully to a multi-attribute utility model, an attribute must have the following properties: first, it should be so defined that all options can be rated with respect to that attribute such that an option with a high rating would be preferable to one with a lower rating on the same attribute, assuming all other factors were equal; and second, at least two of the options must receive different ratings on that attribute. (Sometimes, it is desirable to include a factor on which no difference is observed, even though it does not contribute to the evaluation itself, simply to document the judgment that no significant differences exist; however, such factors can be added at the end of the analysis, for purposes of face validity only, and need not appear until then.)

The task of attribute identification is simply to list a number of qualitative areas of difference among the options (i.e., a number of attributes), and for each area, to contrast the characteristics of the option with those of a less desirable one. In other words, each attribute will be characterized by an attribute name, a description of the positive direction, and a contrasting description of the negative direction.

Current procedure. The present version of QVAL begins by assuming that the user can think of several attributes without detailed prompting (after seeing a few examples of the kind of definition required). Therefore, it simply requests the user to complete a number of sentences of the following type: "A desirable option \_\_\_\_\_, whereas a less desirable option \_\_\_\_\_. (Attribute name: \_\_\_\_\_)" (For example, the blanks might be filled by "is inexpensive," "is more expensive," and "Price," respectively.) The user determines when to stop naming new attributes simply by responding to a yes-no question after entering each new attribute name. (Note that although the attribute name is used for convenience throughout the program, it is really

only a shorthand abbreviation for the dualistic contrast between the positive and negative directions which actually define the attribute.) The final result of the current attribute identification routine, then, is a list of attribute names, each associated with a description of the corresponding positive and negative directions.

Theoretical analysis. The current procedure assumes as its foundation that the user knows enough about the generic class of options being considered to be able to specify a few meaningful attributes without explicitly referring to the specific options under scrutiny. In terms of speed, it cannot be surpassed without sacrificing some essential information; and if the user is reasonably familiar with the space of options being considered, it will result in a fairly complete model of the major utility attributes.

However, a number of potential problems can arise using a simple method like this one, even as a "first cut." If, for some reason, the options being evaluated do not fit into any readily identifiable category which can be exploited to identify utility attributes, the user may find himself strained by what may turn out to be inappropriate efforts. Furthermore, an overambitious user might think of an extremely long list of overly specific utility attributes (for example, analyzing a system into components at a very microscopic level). Finally, there might be a tendency, whether conscious or unconscious, to bias the list of factors by recalling several highly similar attributes while perhaps neglecting equally important ones that could counterbalance them (but might seem inconsistent).

For the most part, these problems are unlikely to be serious, particularly as there are several opportunities to correct them later in the analysis. In addition, since we have assumed at least some informal discussion before the



program is invoked, it is reasonable to expect the user to have a modicum of familiarity with the option set. Nonetheless, it might well be necessary to constrain the user in the future, to prevent some of the excesses which might innocently arise.

Patrick Humphreys' MAUD program (Humphreys and Wisudha, 1979) has also addressed the issue of attribute identification, but in a somewhat different way: whereas the present program begins by focusing on the generic set of options, and defines preferences which apply to all options in the generic category, MAUD focuses on triads of specific options, in an attempt to discern patterns of similarity and difference. Based on a theory of "constructs" which was developed by the personality psychologist George Kelly (1963), Humphreys presents a triad (say, A, B, and C) of options, asking whether one of them differs from the other two. Now, if, for example, C is said to differ from A and B, MAUD follows with a sentence completion question of the form, "C is \_\_\_\_\_; on the other hand, A and B are \_\_\_\_\_." Having elicited the names of two "poles" (which are somewhat like the "directions" in the current QVAL model), MAUD then scores the remaining options on a seven-point scale. A major point of departure, however, is that MAUD's scales need not be monotonic (e.g., strictly increasing) in terms of value: the "ideal" value may lie somewhere between the two extremes. Thus, in order to transform its scales into utility components, MAUD must "unfold" the scale about its ideal point, a procedure that involves several scaling assumptions which might be questioned; and in any event, the utility attributes thus defined do not incorporate readily identifiable directions.

In general, if any attribute achieves its best value near the middle of its scale, rather than increasing throughout, there has probably been insufficient analysis: the

reason why the utility curve rises as it approaches the "ideal" point is not the same as the reason it falls off after it passes that point. Instead, there are actually two or more factors operating on what happens to be the same range of values, such that one factor dominates in one part of the range, and the other dominates elsewhere. For example, an automobile driver may determine an optimal speed which is short of the car's maximum achievable rate. The reason for not going slower involves the desire to get somewhere as quickly as possible; but the reason for not going faster is a qualitatively different desire to avoid a traffic ticket or an accident.

In general, MAUD's elicitation procedure rests on the construction of a rating scale based on "objective," or directly measurable, quantities, with the elicited scores transformed into monotonic utility ratings by comparison with an indicated "ideal point." QVAL, on the other hand, addresses the question of utility directly, restricting the user to attributes which are necessarily monotonic in value; the user must, if faced with a non-monotonic dimension, either subdivide it into factors which are themselves monotonic, or else intuitively determine the degree to which each item deviates from the "ideal." Thus, while QVAL is faster and less sensitive to the exact location or definition of the "ideal point," the more indirect procedure has the advantage of publicly displaying the "objective" data and the location of the "ideal." These two approaches might be effectively reconciled by eliciting both "objective" ratings and utility scores, and then checking for consistency between the two methods (if the gain in accuracy arising from such a procedure would outweigh the additional investment of time, training, and complexity).

Recommended future development. A desirable "compromise" between the current QVAL method for identifying

attributes and that used by MAUD would begin by querying the user about his state of familiarity with the relevant attributes. A user who could easily generate a few meaningful utility attributes could begin immediately with a QVAL-type elicitation, while one with a less definite idea could invoke a more instructive, MAUD-like routine, which would incorporate more testing for monotonicity, independence, and other desirable features of a utility attribute. This approach is in keeping with a general sense that any interactive decision aid ought to permit users with varying degrees of preparation or familiarity to call upon routines more or less tailored to their particular needs: the novice or infrequent user might require an explicit, step-by-step routine which instructs as it progresses, while the more experienced user would prefer a faster version with more abbreviated inputs and outputs.

### 3.4 Preliminary Elicitation of Scores

Task description. Having identified a set of options and a preliminary set of utility attributes, the user must now enter a set of ratings corresponding to the relative scores of the various options on each attribute. For the purpose of an additive multi-attribute utility model, it is necessary to specify not only the order of preference among the options, but also the relative degrees of preference among alternatives. The necessary information elicited during this step will specify which option is best with respect to each attribute, which option is worst, and where the other options should fall on a linear scale between the two extremes. (Although it might be possible, there is no requirement at this stage to estimate the absolute magnitude of any preference, nor is there a need to compare scores across attributes; those judgments will be required only at a later stage.)

Current procedure. After reminding the user of the positive and negative directions which define a given attribute, QVAL lists the options (each with an associated index number), and asks the user to specify the best and worst of those outcomes with respect to the given attribute. Having identified these two extremes, QVAL proceeds to display a graphical 0-to-100 scale, indicating the worst option (by number) at the 0 point, and the best at the 100 point. The user is then requested to type the index numbers of the remaining options below the scale, in positions corresponding to their relative values with respect to the two endpoints. This process provides the user with feedback in two formats--the graphical, analog display on the scale, and a numerical, digital summary of the implied scores--which the user may accept immediately, or adjust. This procedure occurs once for each of the attributes defined in the attribute identification routine.

Theoretical analysis. Methods of eliciting scores can differ in three major ways:

- o the kind of measurement scale requested;
- o the elicitation questions which can provide the required information; and
- o the format in which such elicitations (and the resulting information) are presented and displayed to the user.

As specified in the task description, the minimum information required of the user is a set of relative scores for the options, on an interval scale. QVAL requests only this information, although some circumstances could be envisioned where (for reasons peripheral to the actual decision) additional information might be desired, and available at little additional cost.

One possibility would be to construct an extended scale, specifying two fictitious endpoints: a null option, which would provide absolutely no benefits; and a gold-plated option which would combine or surpass the best aspects of all options. It should be emphasized that these endpoints will, in general, be not only fictitious, but actually impossible to realize under any normal circumstances. Thus, to use such a scale would require a far more complicated explanation to the naive user, and might still result in communication difficulties. On the positive side, however, an extended scale of the type envisioned would allow the user to rate options on a ratio scale, permitting judgments such as "option X is twice as valuable as option Y" to have meaningful interpretations. Furthermore, if the gold-plated option can be defined as the absolute ideal (i.e., the performance of an option unconstrained by cost, information, or other limitations), then it becomes possible to rate the options on an absolute scale, in terms of what fraction of the total possible benefit they provide.

Because of QVAL's primary orientation towards rapid structuring and decision making, the additional time required to construct ratio or absolute scales would be an unnecessary burden to the current system. However, if in the future it became desirable to incorporate such an optional feature, the benefits might be worthwhile. For example, in the event of a large number of possible options to choose from, an absolute scale allows the options to be considered separately, in a manner whereby they can still be compared with one another (relative, interval scaling requires the explicit comparison of all options with one another, rather than with the absolute endpoints).

In addition to the kind of scale obtained, decision aids might also vary in the way of eliciting scaling information. For example, the judgments can be direct magnitudes,

rankings with subsequent estimation of differences, conjoint scaling comparisons, or any of a wide variety of others. In theory, if a subject's utility scale is precise, constant, and perfectly known, all methods of elicitation should yield the same results; however, imprecision, lability, and imperfect knowledge combine with the noise and biases inherent in any subjective measurement procedure to produce results which might be incoherent.

One way of examining various elicitation methods is to determine the possible sources, magnitudes, and directions of elicitation error. The results of such analysis would permit the analyst to select the single procedure with the best overall performance or, given sufficient time and resources, the combination of procedures which could "triangulate" to a more accurate measurement. Again, because of its critical constraint on operation time, QVAL should probably restrict its operation to a single method, which in the current system is direct rating, modified by the prior selection of a best and worst value.

Recommended future development. Ideally, a single routine could be developed that would determine information about the scores on any given attribute with as few questions as possible at a given level of accuracy. For example, the options could be rank-ordered first, and then the intervals between adjacent options on the ranking scale could be assigned proportional values, etc. Much of this could be greatly simplified by a more sophisticated interactive graphics capability than is available with the current QVAL as implemented on the IBM 5110 computer.

As a potential improvement, the user could be queried as to whether or not he wished to include an "absolute zero," a "status quo" option, or a "gold-plated" option along with the true options; if he did, the appropriate

scale would be constructed and the remainder of the options could be evaluated in terms of those "baselines."

### 3.5 Assignment of Attribute Weights

Task description. In order to combine ratings on one attribute with other attributes, it is necessary to elicit a weight for each attribute, such that the ratio between any two attribute weights indicates the relative impact of a given point difference on each of the two attributes. For example, if Attribute X has a weight three times that of Attribute Y, then a ten-point difference in scores on Attribute X will have the same impact on overall utility as a thirty-point difference on Attribute Y. Because QVAL assumes an additive utility model, these weights are independent of the actual levels on each attribute, so that any ten-point range on Attribute X will be equivalent in impact to any thirty-point range on Attribute Y.

Although weights are traditionally "normalized" (i.e., proportionally adjusted to add up to 100%), the only important information is contained in their ratios. Therefore, it is required only to obtain a set of numbers reflecting the ratios among the weights.

Attribute weights, particularly in relative scoring systems, such as the one used in QVAL, are often confusing to the untrained user. A frequent error is to assign weights according to some absolute intuitive notion of the "importance" of a given attribute, rather than to the relative impact of a specifically defined interval on that attribute. A rough notion of the true meaning of an attribute's weight involves a combination of three factors: (1) the "importance" of the attribute itself (would a noticeable difference on the attribute have a significant effect on overall value?); (2) the "range" of values on the attribute (is there really



a noticeable difference between the best and the worst options on the attribute?); and (3) "applicability" of the attribute (what percent of the time, or in what percent of the cases envisioned, will this attribute be relevant to a preference?).

While it may not be correct to elicit these three values separately and then multiply them, some similar process should notionally guide the judgmental assignment of weights to the attributes.

Current procedure. The current mode of operation under QVAL involves three steps: first, a textual review of the theoretical considerations involved in weight assignments, with an example; next, the selection of the single attribute with the highest weight (arbitrarily assigned a weight of 100); and then, a direct assessment of the weight for each other attribute, proportional to the 100 assigned to the most highly weighted (N.B.: for want of a better term, the word "important" may be used by the analyst to mean "highly weighted," entailing the triple concept of importance/range/applicability outlined above).

Once all attributes have been assigned weights, QVAL transforms them proportionally into normalized weights which add to 100%, and requests the user's confirmation of the rescaled numbers (if the user is unsatisfied, he may specify new values, which will in turn be renormalized for confirmation, until a satisfactory set of weights is arrived at).

Theoretical analysis. In most decision analysis involving multi-attribute utility functions, and particularly in the case of relative scoring systems, the proper definition, elicitation, and interpretation of attribute weights is the most difficult and most easily misunderstood task. The user is being asked to make a very complex trade-off of the following type:

Consider the difference in overall value which would result from changing performance on Attribute X from the level attributed to Option 1, to that attributed to Option 2; now, consider the difference in overall value which would result from changing performance on Attribute Y from the level attributed to Option 3, to that attributed to Option 4; what is the ratio of the difference with respect to Attribute X to the difference with respect to Attribute Y?

Apart from the general confusion occasioned by having to detach single attribute values from four different options (some of which may, in fact, coincide), and to compare the specific intervals in what may, in fact, not be perfectly specified or well-defined values, the user also faces the cognitive difficulty of integrating the three components of weight (importance, range, and applicability) to arrive at a single value.

While QVAL currently requires the user to perform all of the operations described above in a single step, reporting only the final answer, other methods for eliciting such an answer are possible. We shall discuss two of those methods below: the one used in Humphreys' MAUD, which is based on "reference lotteries" which provide a utility scale consistent with the von Neumann-Morganstern definition of utility; and an approximate method which may be used to arrive at a better set of assessments by decomposing weight elicitation into its three components.

MAUD presents the user with two "reference options" (not to be confused with the actual options being analyzed) depicted in Figure 3-1. The first is a "sure-thing" consequence combining a high value on one attribute with a low value on another. The second "option" is a lottery with two possible consequences: either the low value on each of the two attributes, or the high value on both. If the chance of the better consequence on the second option were 100%, the user should always prefer that option, while if the chance

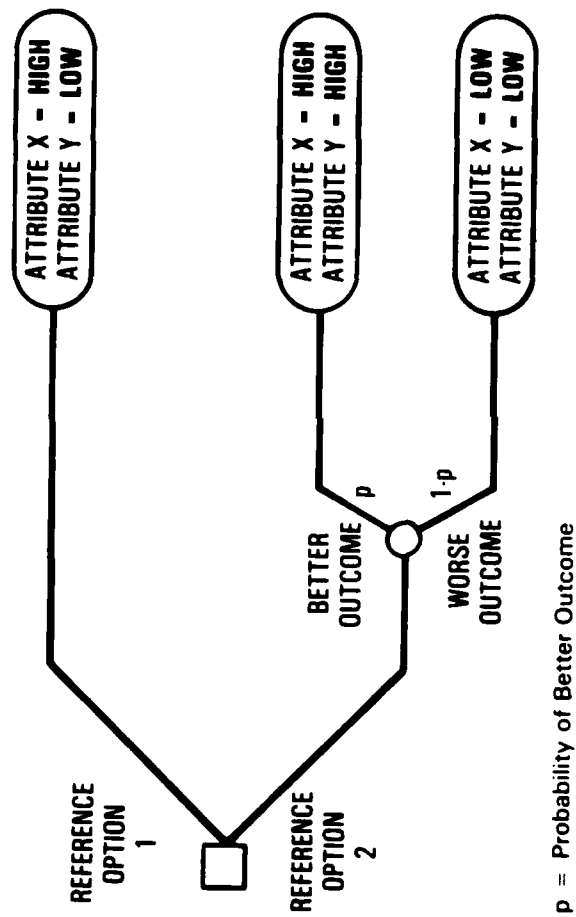


Figure 3-1

# MAUD PROCEDURE FOR ELICITING ATTRIBUTE WEIGHTS

000000011

of the better outcome were 0%, the user should prefer the first option. By asking a sequence of questions varying the probability from 0 to 100% in 10% increments, MAUD determines at what percentage value  $p$  the user's preference shifts from the first to the second option. The ratio  $p:(100-p)$  is then (approximately) equal to the ratio between the weights (assuming the low and high values are in fact the extremes on the two attributes).

While this technique conforms more closely than QVAL's to the basic operational definitions of utility developed by von Neumann and Morgenstern (1953), it will tend to confuse the naive user because of the additional problem of dealing with probabilistic decisions in a situation where such decisions need not be considered. (If, on the other hand, the true options were risky in the sense that their values depended critically on uncertain events, a full-scale decision analysis should be performed, incorporating the user's attitude towards risk in the decision; such a condition, however, would violate the assumption of non-risky alternatives which underlies QVAL and MAUD.) In general, the experience of decision analysts at DDI has been that probabilities and their impacts on perceived value are more confusing to naive users than their potential benefits could justify in non-risky utility analysis.

A different method of simplifying weight elicitation would be to break the concept of attribute weight into its three components (importance, range, and applicability), elicit each component separately, and then combine the elicited values according to a mathematical formula. Such a method involves certain additional assumptions concerning the nature of perceived value on the various attributes. In particular, it involves defining a standard unit of measurement on each utility attribute, independent of the specific options being evaluated, such that a unit increment on any

attribute's utility scale represents a difference comparable in magnitude (not necessarily equal in value, however) to a unit increment on any other attribute. For example, a "unit increment" in the weight of an automobile might be an increase of 5%, while a "unit increment" in gas mileage might correspond to 15%. In other words, a vehicle weight of 2625 would be perceived to be as different from a vehicle weight of 2500 (a 5% increment) as a mileage rating of 34.5 would be from 30.0 (15% increment).

Once the concept of a unit increment has been explained, the user may be asked three relatively simple questions:

1. What is the relative importance (value) of a "unit increment" on each of the utility attributes?
2. How many "unit increments" are included in the range between the worst and the best options on each attribute?
3. If the given utility attribute will be relevant in only certain circumstances (and this partial relevance has not already been incorporated into judgments of importance), in what fraction of the cases envisioned will the attribute be relevant?

It is not in general necessary for the user to specify and precisely define the "unit increment" on each attribute in order to apply the above procedure. The user must be able to conceive of such a unit and be able to compare it with other units; and as long as this can be done consistently, no such specification is needed (of course, for documentation and rationale, such explicit definition might prove highly valuable).

The advantage of the procedure outlined here is that the answer to Question 1 provides a fair approximation to the "absolute importance" of the given attribute, as perceived by the user, while Question 2 captures the completely separate issue of how much the options really differ on the attribute, and Question 3 adjusts the importance (which was assessed under the assumption that the attribute was in fact relevant) to compensate for the frequency, extensiveness, or probability of a given attribute's being relevant.

As a practical matter, Question 1 could be answered by assigning an importance of 100 to the attribute whose "unit increment" was most highly valued and then rating the other attributes proportional to that value. Question 2 is simply a matter of counting units, beginning with the lowest value and imposing unit increments one after another until the highest value among the options is obtained (of course, this procedure will ordinarily take place tacitly, as part of the user's intuitive judgment). Question 3 simply asks for a percentage which (if less than 100) will be used to downgrade the number obtained in Question 1. These three numbers can then be multiplied and the resulting values normalized to determine weights.

The advantage of the above procedure is that the unit increments may be more familiar to the user than the specific options being discussed and may therefore result in a more accurate representation of the user's true values. Meanwhile, the measurement of a range within each attribute can be separated from the rest of the analysis and revised as more data are collected or as options are better defined. Thus, the questions of perceived difference and of perceived value are kept separate, the first reflecting value-free factual information and the second reflecting values independent of the specific options being evaluated.

While it seems that a procedure like the one outlined above would be useful in a routine such as QVAL, some empirical testing with decision-analytically naive subjects should be performed before implementing it. However, intuitive analysis of the underlying reasoning processes and informal preliminary testing have indicated that subjects are likely to feel that the three components are easier to comprehend and closer to a "natural" way of looking at the attributes than a single measure of "weight" as it appears in relative utility modeling.

Recommended future development. For the reasons mentioned in the Theoretical Analysis, it seems likely that the three-component process of weight elicitation described in the preceding paragraphs will prove highly useful in rendering the assessment of weights quicker, less painful, and more accurate as a representation of the user's true values. After some testing and evaluation, a routine to perform the three-way weight elicitation should be added to QVAL, either to replace the existing routine or to provide a choice of methods. In the latter case, the choice could be left entirely to the user, or could itself depend on the answers to questions about the user's decision-analytic knowledge, his specific knowledge about the options and attributes being discussed, and his preference for wholistic versus decomposed assessment techniques.

### 3.6 Calculations and Summary Display

Task description. Once scores have been assigned and weights assessed for all attributes, it is possible to compute the overall score for each option by taking a weighted average of its scores on all the attributes. The information leading to those scores and the final scores themselves can be summarized in a table, schematically designed as follows:



A T T R I B U T E	O P T I O N  1	O P T I O N  2	O P T I O N  3	O P T I O N  4	O P T I O N  5	O P T I O N  6	W E I G H T S
A  B  C  D  E  F  G  H							
TOTAL	TOTAL SCORES						

If the fastest possible decision is required, it will be sufficient simply to select the option with the highest overall score. However, most of the time it is important for the user to examine the display, changing scores or weights after further reflection or upon receipt of additional information, and perhaps adding or redefining attributes to obtain a fuller picture of the values of the options. The intermediate display should be as helpful as possible in guiding this effort to improve the quality of the model and the related assessments.

Current procedure. The existing QVAL has a display as described above, indicating the scores for each option on each attribute, the attribute weights, and the overall scores for each option.

Theoretical analysis. In addition to the information summarized in the table currently displayed, it would be useful for the user to see additional information which might lead to the generation (or recollection) of more attributes, or to the redefinition or reorganization of existing ones. For example, a display of the intercorrelations among the scores assigned to each pair of options might indicate the degree to which the two options in a pair are similar in value (assuming the scores to be accurate); any discrepancy between the user's intuition and these scores should lead to questions which might generate additional attributes to distinguish the options.

Another possibly helpful routine might cluster the attributes according to patterns detected among the option scores. For example, one class of attributes might consist of those on which Option 1 and Option 2 are highest, while Option 5 is lowest, while another category might include those attributes in which Options 1 and 5 are approximately equal, and both are better than Option 3. (Factor analysis or some similar technique could provide such clusters automatically.) The categories thus derived might suggest some underlying property or feature which might lead to the generation of new attributes or the improvement of existing ones.

Recommended future development. Although the simple matrix representation of the scores, weights, and totals is sufficient to capture all of the information which has gone into the decision model, some partial additional processing of that information would make a display more useful. A graphical representation would make the implications of the data much more immediately visible, for example.

It would also be highly desirable to add optional statistical data analysis routines such as the factor analysis

and correlation features discussed above, plus any others that might help guide the user towards a better understanding of the implications of the data he has just provided.

### 3.7 Iterative Expansion of the Model

The attributes selected in the initial modeling stage probably account for most of the major factors which determine an overall preference for one of the options over the rest. However, except in the most time-constrained circumstances, it would be desirable to test the "current best option" to see if, in fact, there might be other attributes, previously ignored, that might reverse the order of preference in favor of another choice. Even if no such reversals occur, it might be desirable (either for the "face validity" that stems from a fairly exhaustive list of attributes, or in order to select a second-best or third-best alternative) to continue probing for new attributes which distinguish among the lesser-valued options. While the processes involved in these two types of expansion (i.e., those which might result in unseating the "current best option" and those which merely affect the magnitude and order of preference among the lesser options) are fundamentally the same, they suggest enough difference to merit separate discussion below.

#### 3.7.1 Additional attributes affecting the best option -

Task description. Assuming the data in the current model are factually accurate, the main danger in simply selecting the option with the highest aggregate score comes from the possibility that some other attribute or attributes exist which, if considered, would lead to a shift in preference, in favor of some other option. While it is impossible for the computerized decision aid (or, for that matter, the analyst without extensive substantive experience

in the field of application) to suggest possible attributes, it is possible to probe the user by focusing on the "current best option" and asking the user specifically to think of new attributes on which that option is not the best one. If such an attribute is found, it will be added to the list of attributes, scored and weighted, and a new overall set of total scores produced, possibly with a different "best option." Of course, if the user cannot think of any such attribute, there is no guarantee that none exists, but a thoughtful search by a well-informed user should turn up virtually all important attributes of this type.

Whenever the process of looking for attributes that place the "best option" at a disadvantage fails to find additional attributes, the presumption is that the "current best attribute" is truly the one which should be selected. At this point, the augmented list of attributes, scores, and weights should be presented for final inspection.

Current procedure. Currently, when QVAL is instructed to elicit additional attributes, a listing of the options and their overall scores, in order from best to worst, appears. QVAL then asks if there is any attribute other than the ones already considered on which some other option is better than the "current best option."

If the answer is "yes," then QVAL asks for the name of that option, the positive and negative directions, the scores for the various options, and the weight to be assigned to the new attribute (proportional to the weights on the other previously defined attributes, in much the same manner as the original elicitations). Once all new elicitations have been completed, new overall values are calculated to incorporate the new attribute, and the process repeats, beginning with the presentation of the options, together with their new overall scores.

When a "no" response is encountered, the elicitation of new attributes terminates, and a new table is displayed, indicating the options, attributes, scores, weights, and overall values, in much the same way as described in Section 3.6.

Theoretical analysis. The current mode of QVAL operation assumes that the only substantive knowledge about the options being evaluated resides with the user. (GenTree, discussed in Section 6.0, represents a proposed concept for a different kind of decision aid, using substantive knowledge which has been stored in a database to guide the generation of attributes; however, this same capability could not be incorporated into QVAL without substantially altering its definition.) Because only the user can provide new attribute names, QVAL must restrict its efforts to prompting the user to think of possible additions to the attribute list.

In addition to the current method (which is to ask what attributes, if any, might counteract the apparent preference for the "current best option"), a number of other possible questions might prove useful. For example, if the sets of scores on two options are highly correlated, it might make sense to ask in what ways, if any, they differ. Or, if one option seems to be completely dominated (i.e., inferior or equal to some other option on every current attribute), it might be worthwhile to ask if any attribute exists which could favor the dominated attribute. Naturally, there would be a trade-off in operational value between the time taken by such questions and their probable value in unearthing new attributes. Perhaps some user guidance could be obtained by assessing an index of how comfortable the user is with the current analysis, and asking further probing questions only if something seems to be missing (of course, if time is not critical, the user can simply stop the session and resume later, presumably with time in between to think of additional attributes).

Recommended future development. Unless some user-guided procedure for memory prompting can be devised, it is unlikely that the current method can be significantly improved upon. Naturally, using database-related techniques or artificial-intelligence programs might be of great advantage in the search for additional attributes, but those would both fall beyond the scope of QVAL's definition.

### 3.7.2 Attributes affecting selection of "next-best" options -

Task description. If the only goal of the decision-analytic effort were to determine the single most highly recommended course of action, it would be advisable to terminate the procedure as soon as the user ran out of additional attributes which could possibly unseat the "current best" option. However, there are three possible reasons for continuing the analysis to include factors which may discriminate only among the less desirable options. The first of these is that if, for some reason, the "current best" option becomes infeasible or inappropriate, the user would like to have as complete an analysis as possible of the remaining ones. Another consideration, "face validity," reflects the need to include attributes which, although irrelevant to the selection of the best option, appear to possess enough "a priori" importance to justify their inclusion for the sake of completeness, persuasiveness, and effective communication. Finally, there remains the possibility, however remote, that by focusing only on the lesser options, the user might recall additional attributes that do impact on the evaluation of the first-place option (but were somehow overlooked).

Current procedure. When QVAL enters the "next-best options" mode, it detaches the "current best" option from the list of options, and proceeds to determine the second-best option by using more or less the same procedure

as that described in Section 3.7.1, with the following exception: Because the "best" option has been assumed to receive the highest score on each new attribute (otherwise, the stage outlined in Section 3.7.1 should not have terminated), that option is automatically assigned a score of 100 on any new attributes (therefore, it is necessary to identify only the worst option prior to assigning scores).

Once the set of attributes possibly affecting the selection of a second-best option has been exhausted, QVAL will permit the user to set aside the two top options, and continue discriminating among the potential "next-best" options in search of a third-best option, and so on until the user decides to stop or until the set of options is completely rank-ordered with no attributes remaining which might alter that ordering. Of course, if at any point a new attribute is recalled on which the previously top-rated option or options are in fact surpassed, QVAL must return to the possibility of altering the order among the entire set of options, and the "next-best" sequence is broken.

Theoretical analysis. In searching for additional attributes, whether in the "best" or in the "next-best" mode, the user is trying to brainstorm, to jog his memory for possible attributes which might have previously eluded him. If it were possible to provide the user with an exhaustive checklist of potential attributes, to be included or excluded after an explicit evaluation, there would be no problem; however, in virtually every analysis of complex options, the number of potential attributes is so great as to make exhaustive listing impractical or impossible. A somewhat more workable method would be to apply previously stored information about the set of options to generate a list of likely candidate attributes--this is the philosophy behind GenTree, described in Section 6.0; however, the



assumption that QVAL has no such previously stored knowledge makes this strategy inapplicable within the limits of QVAL.

Therefore, QVAL must be limited to three kinds of procedures: completely content-free attempts to probe the user's memory; general-purpose "questionnaire" checklists which might happen to include a few useful memory prods; and artificial-intelligence routines which, by "conversing" with the user and analyzing his responses, might lead him to uncover some new attributes. Each of these will be discussed below.

The current procedure for eliciting names of additional attributes, whether for the "best" or the "next-best" options, falls into the first category, completely content-free memory aids. Its rationale is that by focusing the user's attention on a specific option or by comparing that option to those which appear to be less highly valued, the decision aid can create associations which might lead to the recall of attributes. A number of other content-free methods are possible (for example, prompting the user with a letter of the alphabet, and asking for an attribute beginning with that letter, etc.), but none seems to merit enough attention to include it in place of (or in addition to) the current method.

General-purpose questionnaire procedures represent a fast, but highly inefficient, way of prompting the user. Because QVAL has no specific knowledge about the content matter being analyzed, there are two possibilities: either ask very general questions in the hope of stimulating some specific thoughts on the part of the user, or ask specific questions, with a low probability of hitting on something meaningful. Probably the best compromise is a questionnaire which begins with broad categories of attributes and progresses towards increasingly specific questions

in the areas which receive promising responses. For example, an early question might ask whether the options differed in terms of their economic impacts on the decision maker, while later questions (assuming economic impacts did seem important) might ask whether the specific effect on international tariff agreements was an attribute, or whether there would be an effect on overall rate of unemployment.

If QVAL were to be used within a highly limited context (such as aircraft selection, or tactical command and control), it would be possible to tailor a set of questions to the specific field, with a reasonable probability of covering the most likely attributes. However, in the general-applications context QVAL was designed for, it seems unrealistic to expect much of an aid to memory without a great deal of wasted time and effort searching blind alleys. Again, a substantial benefit could arise from capitalizing on a pre-stored database, but this would be beyond the scope of QVAL.

The only reasonable practical alternative to "blind" questionnaires without using an extensive database is to adapt an artificial-intelligence routine (such as Weizenbaum's ELIZA [Weizenbaum, 1965]) to "interview" the user, essentially building up a temporary database from the dialogue. Although dialogue methods in artificial intelligence are still developing, it should now be possible, given a machine with enough speed and memory, to design a program utilizing mnemonics and heuristic devices to "think through" the problem with the user, enumerating possible attributes as they are encountered. Naturally, such techniques would represent a rather large investment given our current technology, and would limit QVAL to operation on fairly large systems with the proper higher-level languages implemented. Furthermore, unless a pre-stored database is included, these artificial-intelligence techniques might require a great

deal of interaction before "learning" enough to be of much assistance. As artificial intelligence continues to develop, and methods become faster, more reliable, and more generally available, this alternative will become more practical; until then, however, it seems too costly compared to the expected benefits to attempt to develop or adapt an artificial intelligence routine for QVAL's purposes (although GenTree, with its database to search, might be another matter).

Recommended development. Although it seems advisable to continue monitoring developments in artificial intelligence, the only immediately indicated efforts which might improve QVAL's efficiency at detecting and identifying possible attributes would take the form of additional content-free prompts, or general-purpose questionnaire routines to provide starting points for the user's associative processes. The more limited the context, the more sense it makes to use a questionnaire-type approach; and as a first step towards content-orientation, it would be advisable to develop specific questionnaires to deal with a few recurring problem areas, for testing and demonstration purposes, at least.

### 3.8 Hierarchical Organization of Utility Attributes

Unless a rapid decision must be made, without subsequent review, revision, or justification, it is generally desirable to restructure the set of attributes to form a more logically organized framework for further discussion, analysis, and final presentation. Whenever more than six or seven attributes are present (i.e., in almost any complex problem), the clearest and simplest form of representation is a hierarchical structure, in which the attributes are grouped into clusters (and some of those clusters perhaps grouped into larger clusters, etc.), so that each cluster can be examined separately from the rest of the model, and then re-integrated into the whole model. For example, in evaluating automobiles,

it might be a natural step to consider one cluster of "cost factors" (purchase price, trade-in allowances, financing terms, insurance cost, etc.), one cluster of "performance factors" (acceleration, cornering, braking, etc.), and perhaps other clusters of "comfort," "status," "appearance," etc.

One major benefit of hierarchical organization is its convenience in displaying and discussing results. Limitations on human information processing (see Miller, 1956) make it difficult to deal simultaneously with seven or more items of information unless they are logically structured in some way. Thus, a list of a dozen or more attributes, or a table of scores and weights for such a list, might generate more confusion than insight (particularly if the attributes are presented in order of recall, or in some other arbitrary order). By organizing the attributes into meaningful groups, a hierarchical structure facilitates thinking about them and presenting their scores and implications in an easily understood manner.

The process of organization may also serve as an added memory aid, no matter how thorough the previous efforts were. Once a user sees a number of attributes grouped in a meaningful way (perhaps with an obvious superordinate category to unite them), other potentially important members of the same category might become apparent.

On the other hand, in the event that two or more attributes tend to duplicate or overlap one another in meaning, a hierarchical structure can highlight the duplication by juxtaposing them, and thus lead to a simplification or redefinition, improving the accuracy of the model. Sometimes, a completely new way of subdividing a cluster becomes apparent, providing a clearer or more complete analysis of its major components.

The juxtaposition of conceptually related attributes in a single cluster is of further use in reviewing and revising the scores and weights associated with those attributes. By comparing the scores on two or more analogous attributes, the user may realize that he has been inconsistent, or that perhaps the weights assigned do not reflect his true preferences accurately. Since it is easier to compare items which are conceptually related, the user can generally improve the quality of his assessments by reviewing them in the context of a hierarchical structure.

Finally, in the process of sensitivity analysis (see Section 3.9), it is useful to see what happens if the weight assigned to an entire cluster is altered (while maintaining the proportional weights within the cluster). While a similar test could be performed on nonhierarchical data, the mechanics would be more difficult and the results less clearly represented.

Task description. Initially, the user has an unstructured list of attributes. The task of the decision aid is to help the user organize those attributes into clusters and sub-clusters in a way which meaningfully reflects the relationships among the attributes. The final product of this phase, then, is a tree-structured outline, on which every "node" represents either a single attribute or a cluster of attributes; multiple nesting of such clusters is permissible when the number of attributes and the degree of organization indicate it.

Current procedure. QVAL currently relies on the user's intuitive judgment to provide structural organization. Initially, the user is presented with the entire list of attributes, and (assuming there are more than three) asked if he wishes to subdivide the list (if there are fourteen or more, the list must be subdivided). Assuming the user does

subdivide the list, QVAL will allow him to partition the entire set into categories, subject to the following restrictions:

- o each attribute in the list must appear in one and only one category;
- o there must be at least one category with more than a single attribute in it; and
- o there must be at least two categories.

Once clusters have been formed, QVAL identifies those categories which contain three or more attributes, and again asks the user if he wishes to subdivide each of those categories. This procedure continues until the user has performed all desired subdivisions, at which point the entire hierarchical structure is displayed in a diagrammatic outline form.

Every category that includes two or more attributes can be assigned an aggregate weight which is the sum of the weights on its component attributes. These weights can be of further use in displaying and reviewing the model's implications, as well as in performing sensitivity analysis.

Theoretical analysis. The current procedure depends on the user's intuitive notions of similarity, comparability, and superordinate categories, in order to achieve a logical and meaningful hierarchical organization of the attributes. To the extent that the user understands (or can be quickly taught) what makes a valid structure, the current QVAL method will be not only correct, but also maximally rapid and efficient. On the other hand, in difficult structures, where the "best" hierarchical structure is not necessarily apparent even to an experienced analyst, it is possible for the user to develop a structure which fails to capture the

essential implications of the given model, or to abandon the entire effort out of confusion.

There are two basic methods to help the user structure the set of attributes while reducing the risk of inappropriate structures. The first of these would resemble the current procedure, but would augment the instructions to ensure that the user has understood what criteria ensure sound structure, and to remind him of the considerations involved in clustering as he follows the procedure. The other method would ask the user for less direct information, which might be easier to obtain and more reliable, and which might then serve as the input for a pre-programmed clustering analysis routine which will automatically develop a structure using the elicited information. Because the first method merely involves additional text of a tutorial nature, it will receive little discussion here; this does not, however, imply that such a method is undesirable, as it may in fact be the only feasible way of assuring adequate performance without a great investment of time and effort on further elicitations.

The remainder of this discussion focuses on the second method, examining a variety of methods which might be used to elicit information about the attributes, and to apply algorithmic procedures to that information, resulting in a well-formed hierarchical structure. The indirect clustering methods available vary in two respects: (1) the type of information elicited about the attributes; and (2) the specific mathematical algorithm used to generate a tree structure from the elicited data. A wide variety of numerical algorithms exist (some of which are summarized in Hartigan, 1975) to perform cluster analysis, and it is not essential to evaluate or recommend a specific numerical procedure at this stage; but some discussion of the types of information elicited will illustrate the potential for future development in this phase of the analysis.

A first approach to automatic hierarchical structuring might involve utilizing information already available to QVAL, in the form of the scores on the various attributes. A table of correlations among the scores on various attributes can be constructed, and then a numerical algorithm applied to group together those attributes which are most highly correlated. Although initially attractive, such a procedure would suffer from several flaws which render it unsatisfactory. First, the fact that the scores on two attributes are highly correlated does not guarantee that the two attributes are causally related in any meaningful way, particularly if the number of options is small enough to allow for frequent instances of high correlation due to random coincidence. Conversely, attributes that are logically related may appear to be uncorrelated when restricted to the few options being evaluated; in fact, two closely related quantities might well be negatively correlated (e.g., an item which scores well on "price" is less expensive and, therefore, likely to score poorly on quality-related attributes, even though each quality-related improvement could be associated with a price increment). Finally, there is a systematic bias resulting from the prior elimination of options which are obviously too bad to be competitive with the one being evaluated: the selection process guarantees that among the options which do compete with one another, there must be negative correlations between attributes (because a positive correlation implies that one option will dominate another on both attributes, and then the dominated option would tend to be excluded prior to the analysis). For these and a number of other reasons, it seems preferable to base the hierarchical structuring on additional information elicited from the user, rather than on the correlations among existing scores.

A semi-direct method of clustering requires the construction of a matrix of similarities (or of differences)



among the attributes, based on the user's direct judgment of how closely related each attribute is to each of the others. One way to arrive at such a matrix of values would be to ask the user to estimate judgmentally the correlations that would be observed if scores were assessed over the entire universe of which the specific options being evaluated are only a small sample. This "pseudo-correlation" approach is not, however, the only viable method for grouping attributes together: two attributes should also tend to be grouped together if they are easy to compare and contrast; if they refer to causally related properties; if they represent independent estimates of the same quantity or property; or if they refer to analogous functions. Indeed, it might be possible to elicit a different matrix of similarities for each of the above criteria, and then aggregate them in some pre-defined procedure, although the additional time and effort of elicitation might not be worthwhile.

At any rate, once a matrix of similarities (or differences) has been developed, it is possible to use a variety of algorithms to produce a hierarchical structure. Given any such structure, the user may display it, modify it, attach labels to the clusters, and otherwise adjust it to ensure a successful encoding of his beliefs about the true underlying structure, or he can reject the hierarchical structure, repeating the procedure with different similarity scores or employing the more direct method used in the current procedure.

The final method to be discussed involves an analysis by "semantic features." In this case, instead of eliciting judgments of similarity among attributes, the decision aid will produce a sort of questionnaire about properties of the attributes (which may be answered in a "yes-no" fashion, or as a percentage rating, or similar scale). For example, attributes of an automobile might be classified according to

some of the following possible features: "reflects performance while automobile is in motion"; "reflects monetary considerations"; "involves the electrical system"; or "tends to favor foreign-made automobiles." Almost any set of criteria can be chosen, as long as enough variety is present, but considerations of time and effort suggest limiting the criteria to a fairly small number. As in a number of other phases of decision aiding, there is a definite trade-off between the generality of the questionnaire items and the usefulness of the responses obtained. In the case of hierarchical structuring, it seems almost a necessity either to restrict the application to a fairly narrow subject area, or to incorporate some sort of artificial intelligence routine to limit the field through interaction with the user. Of course, an elicitation sequence roughly comparable to the "repertory grid" used by MAUD might be a compromise in this respect, but even that involves considerably more time and effort than the more direct method mentioned above.

Overall, therefore, it seems that the most promising way of helping the user derive a meaningful hierarchical structure for a list of attributes would be to elicit similarity judgments directly (but with some tutorial guidance) and then use an automatic procedure to provide an initial version of a hierarchy. Then, after inspecting the outcome, the user may be given the opportunity to change some of the assessed values, to alter the structure directly, or to build a different kind of structure using a direct clustering procedure.

Recommended development. After a series of human-engineering experiments to study performance of naive users on the various types of hierarchical clustering methods, one such method should be selected and incorporated into QVAL. At the present time, the semi-direct methods involving user

judgments about similarity seem to be the most likely candidates, although empirical data will be needed to support this conjecture; in any event, the method should utilize information beyond the scores themselves, while still restricting elicitations and time requirements to a reasonable level. Regardless of the method chosen, it seems advisable to retain the direct clustering routine currently implemented in QVAL (or some close variant) as a default in the event that the additional elicitations cannot be completed, or lead to an unsatisfactory structure.

### 3.9 Sensitivity Analysis and Editing

Task description. When a "final" version of the model has been constructed, complete with scores, weights, and hierarchical structure, it is valuable to use some additional time to investigate the effects of possible changes in some of the elicited values on the ratings assigned to the options, and in particular on the choice of a "best" option. One way to perform such a test would be to edit the values (after saving the originals!) and simply compare the new results with the original ones. A more convenient, systematic, and useful method is to perform a "what-if" sensitivity analysis on the elicited values. In general, this involves specifying the final results for each combination of values as the variables vary within the specified ranges. Typically, because of computational and display limitations, only one variable at a time is altered, but the method could be extended to simultaneous variations in several variables.

Current procedure. At present, QVAL includes complete facilities for changing any scores or weights; for adding or deleting options; for adding or deleting attributes; and for restructuring the hierarchical organization. In addition, a sensitivity analysis routine exists to vary the weight assigned to a single attribute, while observing the effect

on the overall scores for all options, and noting in particular any shifts in the most highly preferred option as the attribute weight changes.

Theoretical analysis. Ideally, QVAL should have an expanded editing capability and a wider range of possible sensitivity analysis routines than the current version provides. In editing, it should be possible to revise and augment the hierarchical utility structure that has been built, rather than simply restructuring it "from scratch." The sensitivity analysis phase could incorporate several more sophisticated features, some of which are discussed below.

At present, sensitivity analysis deals only with changes in the weight of a single attribute (while the others remain in constant proportions to one another). It might be valuable to allow the cumulative weight of an entire cluster of attributes to vary (while maintaining proportionality within the cluster, and among the remaining attributes). Alternately, it might be worthwhile to vary the proportional weights within a cluster, while keeping the cumulative weight for that cluster constant. Finally, whenever a number of attributes (whether in the same cluster or not) all depend on the same fact or observation, it might be useful to vary them all simultaneously in the direction indicated by a change in that variable (an example might be a group of attributes that depend on the prevailing interest rates--if the assumed interest rate changes, the weights of the related attributes should all vary in a consistent manner).

In addition to varying the weights assigned to given attributes, it might be useful to alter experimentally the scores assigned to the various options on a single attribute, or to vary the scores for a single option on all attributes

simultaneously. In fact, if the scores could be expressed as intervals over which the score might range, or even as "fuzzy sets" or approximate verbal descriptions, a sophisticated routine could provide a far more robust test of the validity of the obtained results with not much more elicitation effort than the current version of QVAL requires.

One development which would require a higher level of technological sophistication, but might prove the most satisfactory in the long run, would be to construct an interactive graphic facility, using a visual display and a light pen, joy-stick, or pressure-sensitive data tablet to vary any score, weight, or aggregate set of weights as indicated, while retaining the original values until a "replace" instruction is issued. Additional graphic devices for illustrating the results and locating areas of high sensitivity could be added easily once such a facility was developed.

Recommended future development. While a full-blown interactive graphic system might be somewhat ambitious for the immediate future (even though it would be valuable), it seems advisable at least to implement a sensitivity analysis that can vary the weights on entire clusters of attributes, and change the relative weights of attributes within a cluster while keeping the cluster's weight constant. Furthermore, it should be valuable to allow some variation of the assigned scores within an attribute, as part of sensitivity analysis. One possibility might be to let the user specify an error tolerance (e.g., plus or minus five points), and then automatically to indicate every instance where a change of less than that amount on any single score could possibly affect the selection of first few preferred options.

Investigation into a system for verbal or approximate elicitations of scores and weights might prove fruitful, but

at the present state of development, could be incorporated into a decision aid only on an experimental basis. It would, however, be advisable to undertake such a study, because the benefits of a successful implementation could greatly simplify not only sensitivity analysis, but also the entire process of elicitation.

#### 4.0 A USER'S GUIDE TO DDI'S CURRENT VERSION OF QVAL

This section represents a functional description of a program written by DDI personnel to perform those tasks described as "current procedure" in Section 3.0. It is intended to explain the procedures needed to run the QVAL program to a user who is already familiar with the operation of the IBM 5110 portable computer (an introduction to the 5110 appears in Appendix A). No attempt is made to describe the code, or the algorithm used, from a programmer's point of view, nor is any effort made to deal with every possible exceptional circumstance that might arise. However, the present guide should permit the user to develop and exercise his own utility model, allowing for the most likely difficulties.

##### 4.1 Loading and Initial Routine

QVAL was designed to operate on an IBM 5110 portable computer, accompanied by an IBM 5114 disk drive and an IBM 5103 printer. The computer and the disk drive should be turned on, and when the computer's display shows the phrase "CLEAR WS" the QVAL disk should be placed in the disk drive, and ")LOAD QVAL" should be typed on the computer.

When the program has been loaded from the disk, a heading will appear, followed by an instruction to turn the printer on, as shown here:

```
*****
*
*          QVAL - QUICK EVALUATION ROUTINE
*
*****
TURN ON PRINTER. PRESS EXECUTE TO CONTINUE...
```

Because from this point on, the program controls printer output, the printer should be left on throughout the remainder of the program's operation.

The next question is, "DO YOU WANT TO RETRIEVE AN EXISTING MODEL FROM STORAGE?". In response to this, if you will be recovering a previously-built model for display, editing, or further development, type "YES" (or simply "Y," which throughout the program will be treated as equivalent to a "YES" response), and the program will begin the "load model" routine, described in Section 4.2. If, on the other hand, you wish to construct a brand new model, type "NO," or "N" and the program will begin the "new model" routine, described in Section 4.3.

#### 4.2 Load Model Routine

Once this routine has been selected, the program searches the disk for a library of existing models, and lists those models, each indexed by a number, as shown below:

```
INDICATE YOUR SELECTION BY ENTERING ITS ASSOCIATED NUMBER.  
  
MODELS CURRENTLY AVAILABLE:  
1) CARS  
2) HOUSES  
3) TELEVISIONS  
ENTER SELECTION NUMBER: 3
```

The user is then asked to type the number corresponding to the model he wishes to recover. Once a model has been selected and loaded, the program transfers to the "main menu" routine, described in Section 4.4.

#### 4.3 New Model Routine

If the user does not wish to retrieve an existing model from storage, the program assumes that he wants to begin constructing a new model from scratch. The first question in constructing such a model asks the user to name a generic



category which encompasses the options being evaluated, expressed as a plural noun or phrase. Next, a singular form is entered. Finally, the names of the options are entered, one at a time. (Currently, as many as eight options can be specified.) The format for these questions is a sentence completion, with brackets indicating the allowable length of the answers (longer answers will simply be truncated to fit the allowable length). This routine appears as shown below\*:

PLEASE SUPPLY A GENERIC NAME FOR THE ITEMS TO BE EVALUATED:

THE ITEMS TO BE EVALUATED ARE ALL...[TELEVISIONS ]  
ENTER A SINGULAR FORM : THAT IS,  
EACH ITEM WOULD BE REFERRED TO AS A(N)...[TELEVISION ]

NOW ENTER THE NAMES OF THE TELEVISIONS:  
(WHEN YOU HAVE ENTERED ALL TELEVISIONS HIT [EXCITE])

[19 SONY ]  
[21 SONY ]  
[B&W MAGNAVOX]  
[ADVENT ]  
[ ]

Note that the empty set of brackets on the line below "ADVENT" represents the user's message that there are no more options to be entered.

Once the list has been entered, QVAL verifies the option names for completeness and correctness by listing the names and asking the user if he is satisfied with the labels, as shown below:

YOU HAVE SPECIFIED THE FOLLOWING TELEVISIONS :  
1) 19 SONY  
2) 21 SONY  
3) B&W MAGNAVOX  
4) ADVENT  
ARE YOU SATISFIED WITH THESE LABELS? YES

If the user is not satisfied, QVAL repeats the elicitation of option names, until a satisfactory list has been entered.

---

\*NOTE: The brand names used in illustrative examples throughout this report were selected arbitrarily, only to provide realistic objects to demonstrate the evaluation methods herein presented. No actual evaluation study was performed, nor should the hypothetical ratings and results be interpreted as an opinion about, or an endorsement of, any actual product.

Once such a satisfactory list has been approved, QVAL proceeds with three major steps in new model construction:

1. initial elicitation of the list of attributes;
2. assessment of scores for all options on initial attributes; and
3. determination of relative weights for initial attributes.

Each of these will be discussed in turn.

4.3.1 Initial elicitation of attribute names - The first step in eliciting a list of attribute names is a short tutorial introduction, with an example as shown in Figure 4-1. Once the example has been presented, the user is asked to consider the actual options in the same manner, completing phrases as shown below:

```
PLEASE COMPLETE THE PHRASES:  
A DESIRABLE TELEVISION ...HAS A CLEAR PICTURE ]  
  
WHEREAS  
  
AN UNDESIRABLE TELEVISION ...IS BLURRY ]  
  
PLEASE SPECIFY AN ATTRIBUTE NAME: CLARITY ]  
  
CAN YOU THINK OF OTHER ATTRIBUTES ? YES
```

This process continues until the user indicates that he cannot think of additional attributes he would like to specify. At that point, QVAL begins the process of assessing scores for the options on each of the attributes listed.

4.3.2 Assessment of initial scores - The following paragraph introduces the user to the score assessment routine:

PLEASE CONSIDER NOW THE VARIOUS REASONS WHY ONE TELEVISION MIGHT BE PREFERRED TO ANOTHER. FOR EACH SUCH FACTOR, OR ATTRIBUTE, YOU WILL BE ASKED TO SUPPLY THREE PIECES OF INFORMATION:

- A SHORT DESCRIPTION OF A DESIRABLE VALUE,
- A DESCRIPTION OF AN UNDESIRABLE VALUE, AND
- A ONE-WORD LABEL FOR THE ATTRIBUTE.

PRESS EXECUTE TO CONTINUE...

FOR EXAMPLE, IF YOU WERE EVALUATING PAIRS OF SHOES, YOU WOULD COMPLETE THE SENTENCES AS FOLLOWS (USER RESPONSES UNDERLINED):

A DESIRABLE "PAIR OF SHOES"...IS COMFORTABLE

WHEREAS

AN UNDESIRABLE "PAIR OF SHOES"...IS UNCOMFORTABLE

ATTRIBUTE NAME: COMFORT

PRESS EXECUTE TO CONTINUE...

---

A DESIRABLE "PAIR OF SHOES"...IS INEXPENSIVE

WHEREAS

AN UNDESIRABLE "PAIR OF SHOES"...IS EXPENSIVE

ATTRIBUTE NAME: PRICE

PRESS EXECUTE TO CONTINUE...

---

CONTINUING THIS PROCESS, YOU COULD ADD ATTRIBUTES SUCH AS 'STYLE', 'VARIETY' (WITH RESPECT TO YOUR EXISTING WARDROBE), 'DURABILITY', AND SO FORTH.

PRESS EXECUTE TO CONTINUE...

Figure 4-1

NEXT YOU WILL BE ASKED TO RATE THE VARIOUS TELEVISIONS ON EACH OF THE ATTRIBUTES YOU HAVE LISTED. YOU WILL BE ASKED TO IDENTIFY THE BEST (MOST DESIRABLE) AND THE WORST (LEAST DESIRABLE) TELEVISION WITH RESPECT TO EACH ATTRIBUTE, AND THEN TO INDICATE THE RELATIVE POSITIONS OF THE INTERMEDIATE TELEVISIONS ON A 0-10-100 SCALE.

PRESS EXECUTE TO CONTINUE...

For each attribute named in the list, QVAL performs the following steps:

1. it reviews the attribute name, and the positive and negative directions specified;
2. it asks the user to select from a list of the options the best and the worst option, with respect to the given attribute;
3. it presents a 0-to-100-point horizontal scale, with the index number of the worst option below "0" and the index number of the best option below "100"; and
4. it instructs the user to type the remaining index numbers on the line beneath the scale, in positions corresponding to the ratings associated with the related options.

Figure 4-2 illustrates this process for a hypothetical example. A few observations about this procedure:

- o Because of the limited width of the display screen, the scores corresponding to the points on the scale represent only even numbers.

ON THE DIMENSION 'CLARITY',  
 YOU HAVE INDICATED THAT A DESIRABLE TELEVISION  
 'HAS A CLEAR PICTURE' WHEREAS AN UNDESIRABLE  
 TELEVISION 'IS BLURRY'.

PRESS EXECUTE TO CONTINUE...

CONSIDERING ONLY THE FACTOR OF 'CLARITY'  
 (OTHER THINGS BEING EQUAL) PLEASE IDENTIFY THE MOST DESIRABLE  
 AND LEAST DESIRABLE TELEVISIONS FROM THE FOLLOWING LIST:

- 1) 19" SONY
- 2) 21" SONY
- 3) B&W MAGNAVOX
- 4) ADVENT

ENTER THE INDEX NUMBER OF THE BEST TELEVISION: 1  
 ENTER THE INDEX NUMBER OF THE WORST TELEVISION: 4

- 1) 19" SONY
- 2) 21" SONY
- 3) B&W MAGNAVOX
- 4) ADVENT

INDICATE POSITIONS OF THE REMAINING TELEVISIONS ON THE SCALE:

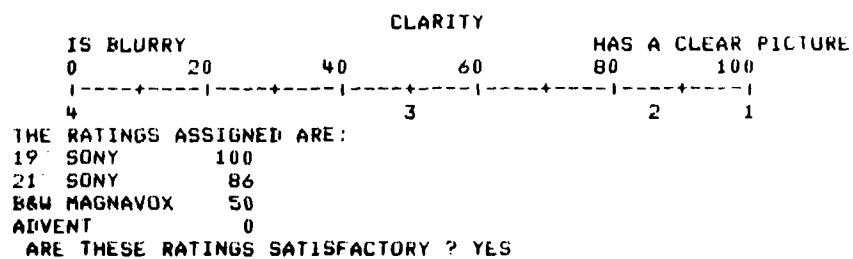


Figure 4-2

- o Currently, there is no provision for assigning exactly equal scores to two or more options in the graphic elicitation (although, as shown below, such scores can be entered numerically as "corrections").
- o Each index number must appear once and only once on the line representing scores (otherwise, QVAL will repeat the elicitation procedure until this condition is satisfied).
- o The "best" and "worst" options must retain their scores of 100 and 0, respectively.

In order to allow the user to correct any typographical errors, to specify equal values for two or more options, or to assign odd-numbered scores, QVAL lists the current ratings and asks the user if they are satisfactory. If the user responds "NO" (or "N"), the scores are listed for editing (which is done by simply changing the line on which the scores are displayed). (See Figure 4-3.) If the scores listed are acceptable, QVAL proceeds to the next attribute; and when all attributes have been scored, QVAL goes on to elicit the relative weights for the given attributes.

4.3.3 Elicitation of relative attribute weights - The weight elicitation routine commences by presenting a brief tutorial introduction, as shown in Figure 4-4. Next, the user is asked to identify the attribute with the highest weight, as shown below:

INDICATE YOUR SELECTION BY ENTERING ITS ASSOCIATED NUMBER.

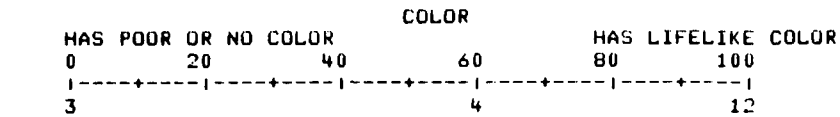
WHICH OF THE FOLLOWING IS OF THE MOST CONCERN TO YOU ?

- 1) CLARITY
- 2) PRICE
- 3) COLOR
- 4) REMOTE CONTROL
- 5) PICTURE SIZE

ENTER SELECTION NUMBER: 1

- 1) 19" SONY
- 2) 21" SONY
- 3) B&W MAGNAVOX
- 4) ADVENT

INDICATE POSITIONS OF THE REMAINING TELEVISIONS ON THE SCALE:



THE RATINGS ASSIGNED ARE:

19 SONY	98
21 SONY	100
B&W MAGNAVOX	0
ADVENT	60

ARE THESE RATINGS SATISFACTORY ? NO

ENTER THE DESIRED RATINGS ALONG 'COLOR' :

				B
				&
				W
1	2	M		
9	1	A		
		G	A	
		N	D	
S	S	A	V	
O	O	V	E	
N	N	O	N	
Y	Y	X	T	
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	
98	100	0	60	(Original Displayed Line)
100	100	0	60	(Same Line, After Editing)

Figure 4-3

THE NEXT PHASE OF THIS ANALYSIS INVOLVES WEIGHTING THE VARIOUS ATTRIBUTE SCORES ACCORDING TO THEIR IMPACT ON OVERALL VALUE.

FOR A GIVEN PAIR OF ATTRIBUTES, YOU WILL BE ASKED TO JUDGE WHETHER THE DIFFERENCE BETWEEN THE BEST AND THE WORST OUTCOME ON THE FIRST ATTRIBUTE IS GREATER THAN, EQUAL TO, OR LESS THAN THE DIFFERENCE BETWEEN WORST AND BEST ON THE SECOND ATTRIBUTE. FURTHERMORE, YOU WILL BE ASKED TO JUDGE THE RATIO OF THE SMALLER DIFFERENCE TO THE LARGER ONE.

---

FOR EXAMPLE, IF YOU WERE CONSIDERING SHOES THAT DIFFERED IN PRICE AND DURABILITY, IT MIGHT BE THE CASE THAT THE PRICES RANGED FROM \$12 TO \$75, WHILE THE DURABILITY RANGED FROM "ABOUT THREE MONTHS' WEAR" TO "TWO YEARS' WEAR". NOW, IF THE \$63 PRICE DIFFERENTIAL MEANT MORE TO YOU THAN THE 21 MONTHS OF ADDITIONAL WEAR, "PRICE" SHOULD RECEIVE A HIGHER WEIGHT THAN "DURABILITY."

---

(NOTE THAT THIS WEIGHT IS NOT MERELY 'IMPORTANCE', BUT THE COMBINATION OF IMPORTANCE WITH A SIGNIFICANT SPREAD BETWEEN THE WORST AND THE BEST VALUES. THUS, AN EXTREMELY IMPORTANT ATTRIBUTE ON WHICH THERE WAS ALMOST NO DIFFERENCE AMONG THE OPTIONS CONSIDERED WOULD STILL RECEIVE A VERY LOW WEIGHT, COMPARED TO A MODERATELY IMPORTANT ATTRIBUTE ON WHICH THERE WAS A GREAT DIFFERENCE AMONG THE OPTIONS.)

Figure 4-4



The most highly weighted attribute is assigned a nominal weight of 100, and then the remaining weights are requested, as proportions of that 100-point weight, as shown:

IF 'CLARITY' IS ASSIGNED A WEIGHT OF 100,  
HOW WOULD YOU WEIGHT THE REMAINING ATTRIBUTES ?  
(A WEIGHT OF 100 MEANS IT IS EQUAL IN IMPORTANCE TO  
'CLARITY'; 50 MEANS IT IS ONE-HALF AS  
IMPORTANT AND 0 MEANS THAT IT IS OF NO CONCERN.)

ENTER WEIGHTS NOW -  
CLARITY                   100.0  
PRICE                    : 40  
COLOR                    : 80  
REMOTE CONTROL           : 5  
PICTURE SIZE             : 15

Once all weights have been entered, QVAL automatically "normalizes" the results, reducing all of the weights proportionally so that the sum of the rescaled weights is 100, as shown here:

THE NEW WEIGHTS ARE THE FOLLOWING:  
1) CLARITY               41.7  
2) PRICE                16.7  
3) COLOR                33.3  
4) REMOTE CONTROL       2.1  
5) PICTURE SIZE         6.3

ARE THESE WEIGHTS SATISFACTORY? YES

A display of the rescaled values is followed by an opportunity for the user to approve or reject the new weights. If in response to the question, "ARE THESE WEIGHTS SATISFACTORY?" the user types "NO" or "N," the weight elicitation is repeated; if the user types "YES" or "Y," the indicated weights are retained, and the program transfers to the "main menu" routine described in Section 4.4.

#### 4.4 Main Menu Routine

Once a model has been loaded from memory, or a new model created from scratch, QVAL enters a phase devoted to

displaying, editing, expanding, structuring, and testing the current model. Access to the relevant program routines is provided through a main menu of choices, which appears initially and then returns whenever the selected routine has terminated. (See Figure 4-5.)

There are seven choices available on the main menu, each of which will be discussed in turn. Briefly, the seven routines can be described as follows:

- |             |  |
|-------------|--|
| EXIT        | QVAL allows the user to store the current model on a disk if desired and then terminates operation.  |
| DISPLAY     | QVAL prints an array with the options and attributes, and all scores, weights and aggregate values, as well as an outline of the hierarchical utility structure, if one has been specified.                |
| MODIFY      | QVAL assists the user in changing current scores, weights, or labels, and in adding or deleting options and attributes.  |
| SENSITIVITY | QVAL permits the user to observe the effect on overall scores as the weight on a specified attribute varies between two given limits.  |
| NEXT-BEST   | QVAL continues probing the user to specify additional attributes that could impact upon the choice of a second-choice, third-choice, etc., option (but will not change the selection of the first choice). |

PLEASE SELECT ONE OF THE FOLLOWING PROCEDURES:

- 1: EXIT FROM QVAL
- 2: DISPLAY RESULTS
- 3: MODIFY SCORES, WEIGHTS, OR MODEL
- 4: SENSITIVITY ANALYSIS
- 5: SELECT NEXT-BEST OPTION
- 6: BUILD HIERARCHICAL STRUCTURE
- 7: ADD MORE DIMENSIONS

PLEASE TYPE THE NUMBER OF THE DESIRED PROCESS: 2

Figure 4-5

BUILD QVAL assists the user in constructing a hierarchical organization for the list of attributes currently specified.

MORE QVAL continues probing the user in search of additional attributes which might affect the selection of a most-highly-preferred option.

By typing the appropriate selection number, the user transfers program control from the main menu to one of the seven routines.

#### 4.5 EXIT Routine

When the user selects the EXIT routine, QVAL first asks if the user wishes to save the current model on the disk. If the user so wishes, the disk is searched and the existing models are listed for the user's information. The user is then requested to specify a name for the model to be stored. This name may be identical to one that appears on the list, or it may be a new model name. If the name matches an existing model name, the user will be asked to verify his intention to replace the old model with that name by the new model being stored. If, on the other hand, a previously unused name is selected, the new model is added to the existing ones (unless the storage capacity of eight models per disk is already filled, in which case the user must either use a different disk, delete one of the earlier models, or abandon his effort to save the model).

Once any desired storage operation has been completed (or immediately, if the model is not to be saved), QVAL prints "PROGRAM ENDED. TO RESTART, TYPE 'RUN' & PRESS EXECUTE" and then stops. Typing "RUN" at this point will start QVAL over, with no model in memory (i.e., exactly as if the computer had just been turned on and the QVAL disk loaded).

#### 4.6 DISPLAY Routine

When the user specifies the DISPLAY routine, QVAL prints a table which lists the names of the options as column headings, the attributes at row headings, the assessed scores in the body of the table, attribute weights in the right-hand column, and aggregate utility scores for each option in the bottom row (Figure 4-6).

				B & U	
		1	2	M	
		9	1	A	
		..	..	G	A
				N	D
				A	V
TELEVISIONS:		S	S	V	E
		O	O	O	N
		N	N	X	T
		Y	Y		
<u>ATTRIBUTES</u>		<u>I</u>	<u>I</u>	<u>I</u>	<u>I</u>
CLARITY		100	86	50	0
PRICE		80	70	100	0
COLOR		100	100	0	60
REMOTE CONTROL		100	100	0	0
PICTURE SIZE		20	40	0	100
TOTALS		92	85	37	26
PRESS EXECUTE TO CONTINUE...					

Figure 4-6

In addition, if a hierarchical structure has already been specified (see Section 4.10 for the description of the BUILD STRUCTURE routine), it is displayed in an outline form, as illustrated in Figure 4-7. The outline structure is indicated both by indentation and by the connecting dotted lines in the display: 'OVERALL' has four components ('VIEWING,' 'PRICE,' 'REMOTE CONTROL,' and 'REPAIR FREQUENCY'); 'VIEWING' in turn has three components, the third of which ('VIDEO QUALITY') is itself analyzed into three components.

#### HIERARCHICAL STRUCTURE

```
OVERALL (100.0)
|---VIEWING ( 79.6 )
|   |---RECEPTION QUALITY ( 8.3*)
|   |---SOUND QUALITY ( 3.7*)
|   |---VIDEO QUALITY ( 67.6 )
|       |---CLARITY ( 34.7*)
|       |---COLOR ( 27.8*)
|       |---PICTURE SIZE ( 5.2*)
|---PRICE ( 13.9*)
|---REMOTE CONTROL ( 1.7*)
|---REPAIR FREQUENCY ( 4.8*)
```

Figure 4-7

In parentheses at the right of each attribute name (or attribute cluster name) is a number in parentheses, indicating the percentage of the overall 100 weighting points accounted for by the given attribute or cluster. An asterisk indicates that the weight is for a single attribute, rather than a higher-order cluster.

Once the value array and the hierarchical structure (if appropriate) have been displayed, control returns to the main menu.

#### 4.7 MODIFY Routine

When the MODIFY routine is selected from the main menu, a secondary menu appears, with eight possible operations listed, as shown here:

PLEASE SELECT FROM AMONG THE FOLLOWING:

- 1: MODIFY WEIGHTS
- 2: MODIFY SCORES
- 3: DISPLAY CURRENT VALUES
- 4: ADD AN OPTION
- 5: DELETE AN OPTION
- 6: DELETE AN ATTRIBUTE
- 7: MODIFY OPTION LABELS
- 8: MODIFY ATTRIBUTE LABELS

ENTER THE NUMBER OF YOUR SELECTION:

Each of the eight choices will be discussed in turn.

4.7.1 MODIFY WEIGHTS - This permits the weights to be changed, one attribute at a time. QVAL displays the current weights for all of the attributes and asks the user to select the weight to be altered. The user indicates the appropriate attribute, and upon request enters the new weight. Because this modification in general will cause the weights to sum to more or less than 100, the weights are then rescaled proportionally to add to exactly 100. Finally, QVAL asks the user to verify the new weights. If the new weights are unsatisfactory, the modification routine is repeated; if they are unsatisfactory, QVAL asks the user to indicate the next weight to be changed (if no further weight changes are desired, the user simply hits EXECUTE instead of entering the selection number of an attribute, and thereby causes the MODIFY secondary menu to reappear).

In the example in Figure 4-8, the user wishes to change the weight attributed to 'CLARITY' from 34.7 to 10. When these values are rescaled, this value of 10 increases to 13.3, and all of the other weights increase in the same proportion. Figure 4-9 shows a display before and after the weight of 'CLARITY' was modified. Note that the overall scores and the weights of the items in the hierarchical structure changed because of the new weight for 'CLARITY'.

4.7.2 MODIFY SCORES - When the user selects MODIFY SCORES, QVAL asks him to specify the attribute whose scores are to be changed (only one attribute's scores can be changed at a time). Once an attribute has been selected, the current scores for all options, with respect to the given attribute, are displayed. By editing the display line, the user can change any or all of the scores.

In order to ensure that the scores on each attribute range from a low of 0 to a high of 100, QVAL must

INDICATE YOUR SELECTION BY ENTERING ITS ASSOCIATED NUMBER.

CURRENT WEIGHTS:

1) CLARITY	34.7
2) PRICE	13.9
3) COLOR	27.8
4) REMOTE CONTROL	1.7
5) PICTURE SIZE	5.2
6) RECEPTION QUALITY	8.3
7) SOUND QUALITY	3.7
8) REPAIR FREQUENCY	4.8

ENTER SELECTION NUMBER: 1

PLEASE ENTER THE NEW WEIGHT: 10

RE-SCALED WEIGHTS:

1) CLARITY	13.3
2) PRICE	18.4
3) COLOR	36.8
4) REMOTE CONTROL	2.3
5) PICTURE SIZE	6.9
6) RECEPTION QUALITY	11.1
7) SOUND QUALITY	4.9
8) REPAIR FREQUENCY	6.3

ARE THESE WEIGHTS SATISFACTORY? YES

Figure 4-8



				B & U	
	1	2	M		
	9	1	A		
			G	A	
TELEVISIONS:	S	S	N	D	
	O	O	V	V	
	N	N	O	E	
	Y	Y	X	N	
				T	
ATTRIBUTES	I	I	I	I	WEIGHTS
CLARITY	100	86	50	0	( 34.7)
PRICE	80	70	100	0	( 13.9)
COLOR	100	100	0	60	( 27.8)
REMOTE CONTROL	100	100	0	0	( 1.7)
PICTURE SIZE	20	40	0	100	( 5.2)
RECEPTION QUALITY	60	80	100	0	( 8.3)
SOUND QUALITY	30	50	0	100	( 3.7)
REPAIR FREQUENCY	100	90	0	10	( 4.8)
TOTALS	87	84	40	26	

Original Display

#### HIERARCHICAL STRUCTURE

```

OVERALL          (100.0)
  |--VIEWING      ( 79.6 )
    |--RECEPTION QUALITY ( 8.3*)
    |--SOUND QUALITY    ( 3.7*)
    |--VIDEO QUALITY    ( 67.6 )
      |--CLARITY        ( 34.7*)
      |--COLOR          ( 27.8*)
      |--PICTURE SIZE    ( 5.2*)
  |--PRICE          ( 13.9*)
  |--REMOTE CONTROL  ( 1.7*)
  |--REPAIR FREQUENCY ( 4.8*)

```

				B & U	
	1	2	M		
	9	1	A		
			G	A	
TELEVISIONS:	S	S	N	D	
	O	O	V	V	
	N	N	O	E	
	Y	Y	X	N	
				T	
ATTRIBUTES	I	I	I	I	WEIGHTS
CLARITY	100	86	50	0	( 13.3)
PRICE	80	70	100	0	( 18.4)
COLOR	100	100	0	60	( 36.8)
REMOTE CONTROL	100	100	0	0	( 2.3)
PICTURE SIZE	20	40	0	100	( 6.9)
RECEPTION QUALITY	60	80	100	0	( 11.1)
SOUND QUALITY	30	50	0	100	( 4.9)
REPAIR FREQUENCY	100	90	0	10	( 6.3)
TOTALS	83	83	36	35	

Display After

Modifying

Weight of "Clarity"

Figure 4-9

occasionally transform the modified scores, according to the following scores:

- o Whenever the lowest of the "raw" modified scores has a value other than 0, subtract that value from each of the scores.
- o Then, if the difference between the highest and the lowest of the "raw" modified scores is anything but 100, multiply each of the adjusted scores by 100, and then divide by that high-to-low difference.

In the examples which appear in Figure 4-10, changing the top score to 150 results in a proportionate reduction of all scores, by a factor of  $100/150$  (or  $2/3$ ). Similarly, changing the value of the lowest score to -20 (or its alternate symbol on the computer's keyboard,  $\bar{2}0$ ) initiates the following transformation: subtract -20 from all scores, and then multiply the results by  $100/120$ , thus raising the score of 60 to 67.

Any time one of these transformations is performed, QVAL automatically adjusts the weight assigned to the given attribute, so that the impact of a specified difference is not altered unless the scores involved are deliberately modified (i.e., so that any rescaling which expands or shrinks the scale on an attribute will be compensated for). QVAL will notify the user when such transformations have been made, by the message, "(ATTRIBUTE WEIGHTS HAVE BEEN RESCALED.)"

After any score modification, QVAL displays the rescaled scores and asks the user to verify that they are satisfactory. If not (as illustrated in Figure 4-10), the original scores are restored, and the modification process is repeated.

	1	2	B & U		
	9	1	M		
			A	A	
			G	D	
	S	S	N	V	
	O	O	A	E	
	N	N	V	N	
	Y	Y	O	T	
	-1	-1	X	-1	
COLOR	90	100	0	60	--- original score
NEW RATINGS:	90	100	0	60	
ARE THESE SCORES SATISFACTORY?					
	N	N	O	N	
	Y	Y	X	T	
	-1	-1	-1	-1	
COLOR	90	100	0	60	--- user-entered new scores
NEW RATINGS:	90	100	0	60	--- re-scaled scores
ARE THESE SCORES SATISFACTORY?					
COLOR	100	150	0	60	--- user-entered new scores
NEW RATINGS:	67	100	0	40	--- re-scaled scores
ARE THESE SCORES SATISFACTORY?					
COLOR	100	100	-20	60	--- user-entered new scores
NEW RATINGS:	100	100	0	67	--- re-scaled scores
ARE THESE SCORES SATISFACTORY?					
COLOR	100	100	-20	60	--- user-entered new scores
NEW RATINGS:	100	100	0	67	--- re-scaled scores
ARE THESE SCORES SATISFACTORY?					
COLOR	100	100	0	60	--- user-entered new scores
NEW RATINGS:	100	100	0	60	--- re-scaled scores
ARE THESE SCORES SATISFACTORY?					
ANY FURTHER SCORE CHANGES?					
(ATTRIBUTE WEIGHTS HAVE BEEN RESCALED.)					
PRESS <u>EXECUTE</u> TO CONTINUE...					

Figure 4-10

4.7.3 DISPLAY CURRENT VALUES - This operation is exactly like the DISPLAY RESULTS routine available from the main menu. For a discussion, see Section 4.6; for illustration, see Figure 4-9.

4.7.4 ADD AN OPTION - This facility permits the user to specify one or more additional options to be evaluated. Because each of the new options must receive a score on every attribute, the same possibility of a value below 0 or above 100 exists as was discussed in Section 4.7.2; when a new option is assigned a score out of the 0-to-100-point range, the same transformation must be used to rescale the scores, and the same compensatory adjustment in the attribute weights must be used.

After specifying the name of the new option, the user will be shown the scores for the current options on one attribute and asked to rate the new option on that attribute. Any necessary rescaling is done; then, if the user verifies that the rescaled scores are satisfactory, attention shifts to the next attribute. When all attributes have been finished, the user may either add yet another option or return to the MODIFY secondary menu. Figure 4-11 illustrates the results of the model after a fifth option has been added.

4.7.5 DELETE AN OPTION - This operation simply asks the user to select which of the current options is to be deleted, and then removes that option from further consideration. If the score on that option to be deleted is 0 or 100 on some attribute, and to delete that value would result in changing the range of scores, the scores for the remaining options will be automatically rescaled to a 0-to-100 range, and the weights adjusted accordingly. When QVAL asks, "WHICH OPTION IS TO BE DELETED?", if the user does not wish to delete an option, he hits EXECUTE rather than entering the index number of one of the options.

				B & W	1 1	
	1 9	2 1	M A G N A V O X	A D V E N T	P O R T A B L E	
TELEVISIONS:	S O N Y	S O N Y				
ATTRIBUTES	I	I	I	I	I	WEIGHTS
CLARITY	100	86	50	0	30	( 10.8)
PRICE	40	35	50	0	100	( 30.0)
COLOR	100	100	0	60	0	( 30.0)
REMOTE CONTROL	100	100	0	0	0	( 1.9)
PICTURE SIZE	47	60	33	100	0	( 8.4)
RECEPTION QUALITY	60	80	100	0	50	( 9.0)
SOUND QUALITY	42	58	17	100	0	( 4.8)
REPAIR FREQUENCY	100	90	0	10	50	( 5.1)
TOTALS	71	71	33	32	40	

Figure 4-11

4.7.6 DELETE AN ATTRIBUTE - This option permits the user to remove from the attribute list the name of an attribute which was entered in error, which has been included as part of another attribute, or which for any other reason is to be eliminated. The operation begins with a list of the current attributes and a request to identify the one to be deleted, as shown (in this example, an illustrative dummy attribute 'XXXX' has been added to the model previously):

INDICATE YOUR SELECTION BY ENTERING ITS ASSOCIATED NUMBER.

WHICH ATTRIBUTE IS TO BE DELETED ?

- 1) CLARITY
- 2) PRICE
- 3) COLOR
- 4) REMOTE CONTROL
- 5) PICTURE SIZE
- 6) RECEPTION QUALITY
- 7) SOUND QUALITY
- 8) REPAIR FREQUENCY
- 9) XXXX

ENTER SELECTION NUMBER: 9

If the user hits EXECUTE without specifying a number, QVAL returns control to the MODIFY secondary menu; otherwise, it attempts to delete the indicated attribute. As a safeguard, before deleting any attribute, QVAL checks to see if the weight on that attribute is equal to zero. If it is zero, QVAL proceeds to delete the attribute, and all scores for that attribute, with no effect on the overall evaluation scores.

If, on the other hand, the attribute in question has a nonzero weight, QVAL first asks the user to verify his intention to delete, as shown here:

ATTRIBUTE 'XXXX' HAS WEIGHT = 4.8  
DO YOU WISH TO CONTINUE THE DELETION? Y

If the user has changed his mind and responds "NO" or "N," the list of attributes appears again, with another request to identify an attribute to be deleted. If the user does wish to delete the given attribute and responds "YES" or "Y," QVAL first resets the weight on that attribute to equal zero (rescaling the remainder, just as in MODIFY WEIGHTS), and proceeds to eliminate the attribute name from the list and the associated scores from the table of scores. Figure 4-12 shows the display of scores for a model on which a hypothetical attribute, "XXXX," had previously been added, and the same model after that attribute had been deleted. Note that the remaining attributes' weights have been increased proportionally to add to 100, and that the elimination of "XXXX" as an attribute has changed the overall scores (but not the component scores) somewhat.

4.7.7 MODIFY OPTION LABELS - This operation lists the option names and asks the user to indicate by number the labels he wishes to change. The old label for that option is displayed, and then the new label is elicited, as shown below:

INDICATE YOUR SELECTION BY ENTERING ITS ASSOCIATED NUMBER.

WHICH LABEL DO YOU WANT TO CHANGE ?

- 1) 19" COLOR TV
- 2) 21" SONY
- 3) B&W MAGNAVOX
- 4) ADVENT

ENTER SELECTION NUMBER: 2

OLD LABEL = 21" SONY

NEW LABEL : [21 COLOR TV]

When no further changes are desired, the user hits EXECUTE instead of entering one of the index numbers when prompted with "ENTER SELECTION NUMBER:". Changing the option label in no way affects any of the numerical model; thus, if there is an actual change of options, rather than a mere cosmetic change in the name of an option, it will still be necessary to revise the scores accordingly.





4.7.8 MODIFY ATTRIBUTE LABELS - This operation performs the same role with respect to attribute labels that MODIFY OPTION LABELS does for the option names and works in exactly the same manner.

#### 4.8 SENSITIVITY ANALYSIS Routine

The SENSITIVITY ANALYSIS routine allows the user to specify an attribute and observe the effect on overall scores as the weight of that attribute takes on a variety of values between two specified endpoints. In the example shown in Figure 4-13, the weight assigned to the attribute 'PRICE' is experimentally varied in five-point increments between 0 and 50; and for each possible value, the overall utility scores for each option are displayed, with an asterisk indicating which option received the highest score. Because the current weight is 13.88, and no value between 0 and 50 alters the selection of the best option, it can be concluded that the results should be highly robust with respect to variations in the importance attributed to 'PRICE'.

The procedure involves the user's selecting the attribute (dimension) to study, and then specifying the minimum and maximum weights to be tested (with the current weight displayed for reference). QVAL then constructs eleven equally spaced values including the minimum, the maximum, and nine intermediate points and displays a table of results as illustrated. When the user is ready to return to the main menu, he simply hits EXECUTE instead of entering the selection number for another attribute. Because SENSITIVITY ANALYSIS is only an investigative technique, no weights are actually changed by this routine; any desired permanent changes must be entered by using the MODIFY routine.

INDICATE YOUR SELECTION BY ENTERING ITS ASSOCIATED NUMBER.

WHICH DIMENSION DO YOU WANT TO VARY?

- 1) CLARITY
- 2) PRICE
- 3) COLOR
- 4) REMOTE CONTROL
- 5) PICTURE SIZE
- 6) RECEPTION QUALITY
- 7) SOUND QUALITY
- 8) REPAIR FREQUENCY

ENTER SELECTION NUMBER: 2

PRICE CURRENT WEIGHT: 13.88

MINIMUM SENS. WEIGHT: 0

MAXIMUM SENS. WEIGHT: 50

PRICE CURRENT WEIGHT: 13.88

			R	
			&	
			W	
	1	2	M	
	9	1	A	
			G	A
			N	D
	S	S	A	V
	O	O	V	E
	N	N	O	N
	Y	Y	X	T
WEIGHT	I	I	I	I
0	88*	86	30	30
5.0	88*	85	33	29
10.0	87*	85	37	27
15.0	87*	84	40	26
20.0	87*	83	44	24
25.0	86*	82	47	23
30.0	86*	81	51	21
35.0	85*	80	54	20
40.0	85*	80	58	18
45.0	85*	79	61	17
50.0	84*	78	65	15

Figure 4-13

#### 4.9 ADD MORE DIMENSIONS and NEXT-BEST OPTION Routines

These two routines perform similar tasks, the only important difference being that "ADD MORE DIMENSIONS" assumes that the newly added attributes are ones on which some other option performs better than the "current best option," while "NEXT-BEST OPTION" assumes that the "current best option" is also best with respect to any newly added attributes.

4.9.1 ADD MORE DIMENSIONS - The "ADD MORE DIMENSIONS" routine is illustrated in Figure 4-14. "ADD MORE DIMENSIONS" begins by listing the options, together with their overall utility scores. QVAL identifies the "current best option" and asks the user whether he can think of any dimension on which some other option does better than the "current best option." If such an attribute can be found, the user enters the name of that new attribute, and then proceeds through the elicitation sequence described below; if no such attribute can be found, the program returns to the main menu.

If the user does specify the name of a new attribute, the next step is for QVAL to ask the user for positive and negative directions (phrases to describe desirable and undesirable outcomes). Next, the best and worst options with respect to the new attribute are identified (just as in the initial scoring of the first set of attributes, except that QVAL will not allow the "current best option" to be designated as the best on the new dimension). Then, just as in the original scoring, a horizontal 0-to-100 scale is displayed, with the best and worst options located in appropriate places on the line beneath. The user must enter the index numbers of the intermediate options in the appropriate position to indicate their scores; the resulting values are then displayed numerically for the user's approval.

ADVENT 26  
B&W MAGNAVOX 37  
21 SONY 85  
19 SONY 92  
'19 SONY' SEEMS TO BE THE MOST HIGHLY VALUED TELEVISION  
THUS FAR.  
CAN YOU THINK OF ANY DIMENSION ON WHICH ANY OTHER  
TELEVISION WOULD BE MORE DESIRABLE ?

WITH RESPECT TO THE ATTRIBUTE 'RECEPTION QUALITY',  
A DESIRABLE TELEVISION...[HAS GOOD RECEPTION ]

AN UNDESIRABLE TELEVISION...HAS POOR RECEPTION 1  
ON THE DIMENSION 'RECEPTION QUALITY',  
YOU HAVE INDICATED THAT A DESIRABLE TELEVISION  
'HAS GOOD RECEPTION' WHEREAS AN UNDESIRABLE  
TELEVISION 'HAS POOR RECEPTION'.

CONSIDERING ONLY THE FACTOR OF 'RECEPTION QUALITY'  
(OTHER THINGS BEING EQUAL) PLEASE IDENTIFY THE MOST DESIRABLE  
AND LEAST DESIRABLE TELEVISIONS FROM THE FOLLOWING LIST:

- ENTER THE INDEX NUMBER OF THE BEST TELEVISION: 3  
ENTER THE INDEX NUMBER OF THE WORST TELEVISION: 4

FREQUENCY	HAS POOR RECEPTION (%)	HAS GOOD RECEPTION (%)
4	10	90
1	40	60
2	10	90
3	10	90

19	SONY	60
21	SONY	80
	B&W MAGNAVOX	100
	ADVENT	0

1) CLARITY	41.7
2) PRICE	16.7
3) COLOR	33.3
4) REMOTE CONTROL	2.1
5) PICTURE SIZE	6.3
6) RECEPTION QUALITY	.0

1) CLARITY	37.9
2) PRICE	15.2
3) COLOR	30.3
4) REMOTE CONTROL	1.9
5) PICTURE SIZE	5.7
6) RECEPTION QUALITY	9.1

Figure 4-14

Once a satisfactory set of scores has been entered, the weights are displayed (with the new attribute initially assigned a weight of 0). Upon request, the user will enter the proper weight for the new attribute, just as in the MODIFY WEIGHTS operation discussed in Section 4.7.1. The new weights will be rescaled to add to 100 and presented for the user's approval. Once a satisfactory new weight has been established, QVAL recalculates the overall utility values, lists the options with their updated utility scores, identifies the new "current best option," and repeats the procedure of adding, scoring, and weighting new options until the user runs out of additional attributes which might influence the selection of a first-choice option.

4.9.2 NEXT-BEST OPTION - The "NEXT-BEST OPTION" routine is illustrated in Figure 4-15. Initially, it is assumed that on all of the attributes elicited in this routine, the "current best option" will have the highest score (because if some other option had a higher score, the "ADD MORE DIMENSIONS" routine would be appropriate). Thus, the "best option" is excluded from consideration temporarily and is automatically assigned the highest score on each attribute added. The immediate goal of "NEXT-BEST OPTION" is to identify attributes which, even though they do not affect the selection of a "best" option, might still be important in determining the second choice.

"NEXT-BEST OPTIONS" operates in much the same manner as "ADD MORE DIMENSIONS." As long as the user can think of additional attributes on which the current second-best option is less highly rated than some other option (not counting the "best" option, which has been excluded from consideration), the attribute name, positive and negative directions, scores, and weights are elicited, just as in "ADD MORE DIMENSIONS," except for the fact that the best option has been predetermined and, therefore, need not be elicited.

EXCLUDED BEST TELEVISIONS:

19" SONY 87

\*\*\*\*\*

REMAINING TELEVISIONS:

21" SONY 84

B&W MAGNAVOX 42

ADVENT 27

'21" SONY' IS THE NEXT-BEST TELEVISION AT THIS POINT.  
CAN YOU THINK OF ANY DIMENSION ON WHICH ANY OTHER  
REMAINING TELEVISION WOULD BE MORE DESIRABLE?

IF SO, PLEASE NAME THAT DIMENSION (IF NOT, HIT EXECUTE):

ATTRIBUTE NAME: [ ]

DO YOU WISH TO CONTINUE NEXT-BEST OPTIONS? Y

EXCLUDED BEST TELEVISIONS:

19" SONY 87

21" SONY 84

\*\*\*\*\*

REMAINING TELEVISIONS:

B&W MAGNAVOX 42

ADVENT 27

'B&W MAGNAVOX' IS THE NEXT-BEST TELEVISION AT THIS POINT.  
CAN YOU THINK OF ANY DIMENSION ON WHICH ANY OTHER  
REMAINING TELEVISION WOULD BE MORE DESIRABLE?

IF SO, PLEASE NAME THAT DIMENSION (IF NOT, HIT EXECUTE):

ATTRIBUTE NAME: [REPAIR FREQUENCY]

WITH RESPECT TO THE ATTRIBUTE 'REPAIR FREQUENCY',  
A DESIRABLE TELEVISION...[REQUIRES FEW REPAIRS]

WHEREAS

AN UNDESIRABLE TELEVISION...[BREAKS DOWN OFTEN ]

ON THE DIMENSION 'REPAIR FREQUENCY',  
YOU HAVE INDICATED THAT A DESIRABLE TELEVISION  
'REQUIRES FEW REPAIRS' WHEREAS AN UNDESIRABLE  
TELEVISION 'BREAKS DOWN OFTEN'.

PRESS EXECUTE TO CONTINUE...

CONSIDERING ONLY THE FACTOR OF 'REPAIR FREQUENCY'  
(OTHER THINGS BEING EQUAL) PLEASE IDENTIFY THE MOST DESIRABLE  
AND LEAST DESIRABLE TELEVISIONS FROM THE FOLLOWING LIST:

- 1) 19" SONY
- 2) 21" SONY
- 3) B&W MAGNAVOX
- 4) ADVENT

'19" SONY' HAS ALREADY BEEN IDENTIFIED AS THE MOST  
DESIRABLE TELEVISION, FOLLOWED BY ANY OTHER EXCLUDED TELEVISIONS  
ENTER THE INDEX NUMBER OF THE WORST TELEVISION: 3

- 1) 19" SONY
- 2) 21" SONY
- 3) B&W MAGNAVOX
- 4) ADVENT

INDICATE POSITIONS OF THE REMAINING TELEVISIONS ON THE SCALE:  
IN THE FOLLOWING ORDER (WORST TO BEST): 3 # 2 1

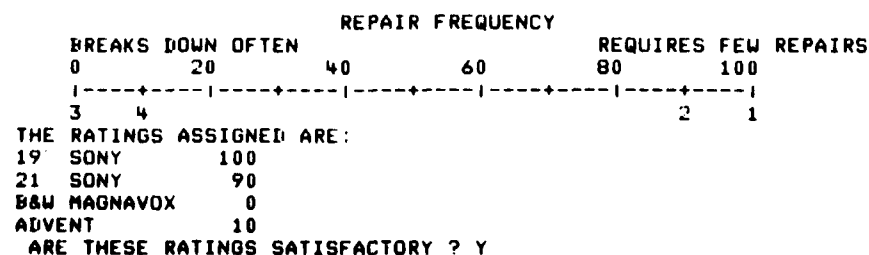


Figure 4-15

When the user cannot think of any more attributes which might alter the selection of a second-best option, he indicates this by hitting EXECUTE instead of specifying another attribute name. Now, QVAL asks, "DO YOU WISH TO CONTINUE NEXT-BEST OPTIONS?" If the user responds "NO" (or "N"), control returns to the main menu level. However, if the user responds "YES" (or "Y"), the elicitation of additional attributes continues as before, except that instead of just one "best option," now the first two choices are assumed to receive the highest two scores (in the same order) on any newly added attributes, and are therefore both excluded for purposes of NEXT-BEST analysis.

When more than one option has been excluded, it can become difficult for the user to recall the order required for the top few scores (which must appear in the proper sequence to avoid contradictions). Therefore, QVAL provides an additional prompt when scores must be entered, in the form of an instruction to indicate the scores on the scale "IN THE FOLLOWING ORDER (WORST TO BEST):", followed by the index number of the option identified as worst, then one or more '#' signs, and then the excluded options, in increasing order of desirability. The only freedom in ordering the index numbers is the order among the "REMAINING OPTIONS" which are better than the lowest-rated one, but worse than the excluded options.

"NEXT-BEST OPTION" continues until either the user indicates a desire to stop (by responding "NO" to the question, "DO YOU WISH TO CONTINUE NEXT-BEST OPTIONS?", or all the options but one have been excluded from consideration. At the end of the routine, the final order among the options is displayed, together with the final set of scores; control then returns to the main menu.

If, in the course of running "NEXT-BEST OPTION," the user discovers an attribute on which the best option

does not in fact surpass all of the others, or if the user finds one in which the excluded options do not receive ratings in the order specified, the proper course is to discontinue "NEXT-BEST OPTION" and return to the main menu. At that point, the user may either run "ADD MORE DIMENSIONS" or return to "NEXT-BEST OPTIONS" without having excluded any but the first-choice option.

There is no restriction on the number of times "ADD MORE DIMENSIONS" or "NEXT-BEST OPTION" can be used. Any time the user has recalled one or more attributes he had previously overlooked, he may (from the main menu) enter the new attributes by using the appropriate routine.

This completes the user's guide to the current implementation of QVAL. In the next chapter, considerations for future implementation will be discussed.



## 5.0 SUMMARY OF THEORETICAL AND METHODOLOGICAL CONSIDERATIONS FOR FUTURE QVAL DEVELOPMENT

This section summarizes those potential improvements in QVAL which involve additional research at a basic level (and therefore have significance beyond the QVAL program itself). Research on some of these topics might be justifiable even without the added benefit of an improved QVAL, because of its impact on the mathematical and psychological foundations of decision analysis and related fields.

### 5.1 Helping the User Generate a Complete, Well-Structured Set of Attributes

Assuming the user is familiar enough with the substantive details of his problem, and that the user's own values (individually or as a proxy for others) are to be applied, the job of eliciting a list of attributes becomes primarily one of aiding the user's recall and eliminating overlap or duplication wherever possible. In the field of psychology of human learning and memory, there are several theoretical results which might point to useful mnemonic devices or strategies for facilitating this recall.

One possible example might be the use of visual imagery--getting the user to draw a diagram of some portion of the options being evaluated, in order to generate additional features which might otherwise pass unnoticed. Another might be to use a variety of "mnemonic templates," easily remembered general-purpose categorization schemes such as the journalist's "Six Honest Serving-men":

I keep six honest serving-men,  
They taught me all I knew;  
Their names are How and Why and When,  
And What and Where and Who.

Cueing recall by arbitrary stimuli (e.g., asking for the name of an attribute beginning with the letter 'T') can also work well, merely by helping the user break free of any "mental blocks" (dead-end patterns of association) which may be inhibiting him. A wide variety of similar schemes are described in James L. Adams' book, Conceptual Blockbusting (1974), and in numerous related works.

In Section 3.0, a few more direct methods for jogging a user's memory were suggested. A list of key words, for example, might be presented, with the user indicating which ones might be appropriate as attributes (with the user's responses possibly being used to select the most likely key words to present next). A given option might be used as a focus, and properties peculiar to that option listed; then, another option could be selected, and so forth. Pairs of options could be selected for contrast, or triads of options could be used as in MAUD (Humphreys and Wisudha, 1979), where the user is asked which two of the three options are most alike, how they are similar, and how the third one differs.

The list of possible mnemonic and heuristic strategies is enormous, and each must be evaluated separately in terms of benefit versus cost. Although additional time and effort might unearth an important attribute that would otherwise have been ignored, there is a diminishing expectation of further discoveries as more time and effort are invested. It would, therefore, be desirable to perform empirical studies to determine the optimum approach before committing to a particular strategy or mix of strategies.

## 5.2 Alternatives to Relative Scoring Methods

The current version of QVAL assumes that options are scored on an interval scale only--that is, the numbers assigned

to the options may all be multiplied by any (positive) constant number, or be increased or decreased by a constant amount, with no loss of information. In other words, the selection of a "zero" point and a unit of measurement is arbitrary. For purposes of standardization, QVAL and several other decision-analytic techniques arbitrarily assign the value of zero to the option which rates the lowest on a given attribute, and establish a unit which is equal to one percent of the magnitude of the difference between the lowest-rated option and the highest-rated one, so that the highest-rated receives a score of 100.

While this method allows for convenient numerical manipulation, restricting all the ratings to the range from 0 to 100, it does present problems in eliciting, interpreting, and communicating a valid set of results. One major conceptual difficulty is that people's prior intuitions about numerical ratings may lead them to the unfounded assumption that a high score must be "good" in some absolute sense, while a low score must be "bad." Similarly, people often assume that a difference of X points on one attribute reflects the same magnitude of improvement as a difference of X points on another dimension.

A related problem is that instead of directly reflecting the "importance" of the various attributes to the user, the weights assigned in a relative model must, if properly assessed, reflect the specific magnitude of value gained by increasing from the lowest to the highest level of performance on each attribute. Unfortunately, in elicitation sessions and interpretive presentations conducted by personnel other than trained decision analysts, this distinction is often forgotten or inconsistently applied. The result is that the weights may reflect either "absolute" importance or the proper relative weight-time-difference measure, may be incorrectly referred to as "importance," and may be assigned numbers which reflect a certain amount of confusion between the two concepts.

One solution to these problems is to resort to an absolute scoring system. Here, the zero and the unit of measurement are absolutely defined: a score of zero represents the complete absence of any performance on the given attribute (or the worst possible level of performance); and the unit of measurement (one point) is defined as one percent of the difference between the zero-level performance and an "ideal" or perfect performance level specified by the substantive expert. For example, in a military evaluation, an absolute zero might refer to a total defeat in a given engagement, while an "ideal" value would be defined as a total victory, with no casualties to friendly troops.

Absolute models are easy to define, but unless the options span almost the entire range of performance, it may be considered meaningless or unrealistic even to discuss the endpoints. Furthermore, if the absolute range is very wide compared with the low-to-high difference among the options, the numerical ratings will be extremely close to one another, usually resulting in a loss of discrimination (e.g., if numerical ratings are given in units of 1/10 point, there are 1001 possible values between 0 and 100, but if the worst of the actual options receives a score of 85 and the best receives a score of 90, there are only eleven possible scores available for intermediate options). Finally, it is not in general an easy task to compare the impact of wide variations in various attributes: the unreality of the zero and the "ideal" points, combined with the likelihood that the assumption of a purely additive utility model will break down (the narrower the range of options, the better an additive utility model approximates any utility function), makes it difficult to assign accurate numerical values to such differences.

A different possible modeling procedure (and one which does not seem to have received much attention in the literature

to date) is the par-value scoring system. Here, an arbitrary value of zero is assigned to some specific level of performance which the user deems representative, or "standard" (often, a "status quo" may be available which is well known to the user); the unit of measurement would be a user-defined quantity representing the smallest "significant" increase (or decrease) from the par value. Scores would then be assigned, using positive numbers to represent the options which represent an improvement over the par value, and negative numbers to represent those options which perform worse than the par value. Weights under this system would reflect the impact on overall value of the "smallest significant difference" on each of the attributes. While the par-value scoring system seems to solve many of the problems which arise from either absolute or relative scoring methods (while sacrificing none of the theoretical soundness), more testing is needed to validate it as an empirically justified procedure.

The three methods described above (relative, absolute, and par-value) all make use of the fairly restrictive assumption that utility scores satisfy the requirements for an interval scale completely. However, it is possible that those requirements may be relaxed considerably without adversely affecting the quality of decision making. Particularly with the emphasis QVAL places upon speed and simplicity, it would be worthwhile to investigate methods (such as conjoint measurement) based on simple fundamental judgments from which one can derive consistent results without relying on the stronger assumptions underlying interval scaling: ordinal ranking and interval comparison, for example. If suitable methods based on these simpler operations could be developed and instituted (capitalizing on the computer's speed of calculation to derive the final numerical values), the benefits would include not only more directly observable justification for the final results, but

also easier elicitation, simpler sensitivity analysis, and freedom from some of the biases which might influence direct numerical judgments of value.

There is still a considerable amount of research, development, and testing needed before any single method of eliciting scores and weights can be certified as the "best" one for QVAL (or for other decision-analytic applications). Because the ultimate success of any decision-analytic model depends critically on the decision maker's ability to define terms precisely, to assign scores and weights consistent with those definitions, and to interpret and communicate results in an understandable and justifiable manner, any new method which facilitates these goals will prove to be a significant advance in the state of the decision-analytic art.

### 5.3 Reducing Biases in Elicitation

As experimental psychologists have long known, almost any elicitation procedure is subject to response biases of various sorts. Some of those biases reflect differential response difficulties (e.g., it is easier to give a verbal response of "twenty-five" than one of "twenty-seven point three six seven nine," even though the second of the two numbers referred to may be closer to the "true" quantity being assessed). Other biases may represent attempts to maintain consistency during a sequence of elicitations without changing previous responses (for example, if one option is scored as 95, and then it turns out that there are two more options which deserve higher ratings, but cannot be rated higher than 100 because of the rating system, the user may bunch the top few scores more closely than his actual judgment would indicate); on the other hand, perhaps in anticipation of such difficulties, subjects may unduly avoid extreme-value judgments, particularly in the early phases of

elicitation. Many of these effects reflect substantial variation among individuals, so that no single general routine could be expected to compensate for any such biases.

A final source of biases in numerical elicitations is the number system itself: having dealt with numbers all their lives, people have developed certain intuitive notions which, while adequate for most situations, are misleading when a particularly accurate numerical assessment is required. For example, even though they might logically agree, given time to think about it, many people do not immediately see that the following sentences are all equivalent:

- o A is 50% more than B.
- o Of their sum, A accounts for 60%, while B accounts for 40%.
- o The difference between B and A is one-third of A.
- o The ratio of A to B is 3:2.

Additional numerical biases result from our unfamiliarity with extremely large or extremely small quantities (such as probabilities of 0.000000065, or consequences billions of times as serious as their alternatives), from our inability to perform rapid complex numerical calculations, and from our pre-conditioned attitudes towards fractions and negative numbers.

While it is impossible to eliminate elicitation biases completely from any system, steps can be taken to reduce or to counteract these biases where they arise. The three major ways of doing this are: (1) to find systems with the smallest possible inherent biases; (2) to use multiple measurements and a variety of different methods to "triangulate," converging

on a single value which, hopefully, allows the biases to cancel one another out; and (3) to measure and calibrate the response biases of individual subjects on similar tasks, and apply specific mathematical corrections to compensate for those biases (of course, verifying the derived final results with the users). Beyond these methods, an adequate sensitivity analysis may determine whether any bias-induced systematic errors could possibly be large enough to alter the implied course of action.

While there are far too many possible elicitation methods to discuss fully in this report, one suggestion which certainly deserves further research is the use of non-numerical methods, and of interactive graphical elicitations (whether performed by the computer or off-line). The development of a sophisticated interactive graphic capability for score and weight elicitation would permit the convenient collection of assessment data, while avoiding some of the response biases inherent in numerical systems and many of the problems arising from sequential elicitations. Naturally, it is possible that other biases might replace the ones avoided, but suitable user-engineering and the convenient accessibility of multiple methods of elicitation would most likely minimize the overall bias involved (and the mixture of numerical and graphical methods could further reduce the impact of any single method's bias).

#### 5.4 Top-Down Versus Bottom-Up Weighting in Hierarchical Utility Models

The current QVAL procedure, because of the sequence in which it operates, must elicit its initial set of weights on an attribute-by-attribute basis (since the hierarchical structure is built afterwards). Once a hierarchy has been developed, the "cumulative weights" assigned to the clusters are computed simply by adding the weights of the component



attributes within each group. If the hierarchy is graphically represented in the standard tree-structure format, with the "overall" utility node at the top and the branches extending downwards, it is clear why this method of weighting is referred to as "bottom-up."

An alternate weighting method, the "top-down" approach, begins with the hierarchical structure and divides the overall utility weight (100%) among the major components represented by those nodes which appear at the first level below "overall." For each first-level node which represents a combination of two or more second-level nodes, the percentage attributed to each of the second-level components is assessed. This top-down process of subdividing weights continues until only single attributes remain.

Although the top-down and the bottom-up approaches should theoretically produce identical results, a systematic bias makes them differ considerably in practice. Subjects in general have a tendency, when directly assigning weights, to avoid an extremely large range (since they rarely wish to assign a weight very close to zero, and may for convenience deal in units of one, five, or even ten percent). This means that in bottom-up weighting, the attributes will tend to be weighted more equally than the facts and the user's values truly justify; therefore, a higher-level cluster of attributes with several components will almost always receive a higher aggregate weight than one with few components. On the other hand, in top-down weighting, the tendency will be for those attributes which are members of large clusters, or which are several levels removed from the "overall" level, to be assigned weights far lower than those attributes which appear near the top, or on nodes with few branches.

With a relative utility model such as the one currently used by QVAL, a top-down approach leads to another, more

serious problem: in order to judge the proper weights on higher-order categories of attributes, subjects must evaluate the simultaneous impact of a variety of improvements, versus a variety of other improvements. For example, the user might have to judge the relative merits of two options: the first with the lowest possible values on six attributes and the highest values on four other attributes; and the second with the highest values on the six attributes and the lowest on the other four. This highly complex task is likely to tax the user's cognitive abilities, leading to simplified heuristic strategies which will, at best, be highly time-consuming and unpleasant and, at worst, lead to incorrect responses. To use a scoring system where the weights correspond directly to "importance" would remove this problem, at least in those models where the clusters of attributes can conveniently be assigned a joint importance.

Current decision-analytic practice solves the problem of conflicting biases in the two elicitation systems by an iterative process whereby the client observes the implications of one method upon the other's results, and continues to adjust the numerical values in both methods until the results converge in a consistent manner. To facilitate this, an interactive computer graphics capability would allow simultaneous observation of the two methods' implications, and provide the user with more rapid control over the variation of values.

Beyond the simple implementation of an iterative adjustment process on a computer, research might also lead to the development of new methods which would eliminate or greatly reduce the kind of bias encountered in weighting. For example, a method might be investigated using an intermediate level of aggregation as a starting point, and then aggregating or subdividing as necessary to achieve a satisfactory model. There is substantial room for the creation of new methods for obtaining correct weights.

### 5.5 Incorporating Non-Additive Interactions into the Models

In the vast majority of multi-attribute evaluations, the well-trained decision analyst can guide the construction of a model in such a way as to minimize the impact of any dependencies or interactions which might lead to a departure from additivity. One potential problem with QVAL or similar aids is that the analyst's experience and intuitions in this matter are extremely difficult to capture, and therefore the possibility of a serious interaction or dependency is greater.

In general, there are three major strategies with respect to interaction: (1) ignore them initially, using sensitivity analysis to test for the impact of any significant interactions and dealing with those after the fact, preferably with a trained analyst's help; (2) restructure the model in such a way that any interactions can be eliminated by redefining attributes, or by combining groups of interacting factors into a single aggregate factor; and (3) providing a first-order multiplicative approximation for certain interactions identified as potentially important.

The first of these options has much to recommend it, and as long as the user can be persuaded to reserve judgment about any possible interactions until the preliminary analysis has been completed, is certainly the simplest and safest method.

Restructuring the model to avoid duplications, dependencies, and other interactions might prove extremely difficult in the absence of a trained analyst (or an extremely sophisticated artificial-intelligence routine). Except for the relatively simple expedient of combining interacting attributes into a single attribute (thereby reducing the complexity of the model, but making the assessment of that attribute more difficult), interactive restructuring methods are likely to

take longer and produce more confusion than their potential benefits would justify.

For simple two-factor interactions, there is a method which can be applied to produce a good first-order multiplicative approximation. For simplicity, assume a 0-to-100 range of scores for the options, as is the case in the current relative-scoring version of QVAL (other scoring methods can be dealt with in an analogous way, simply by transforming the value scales). The assessment of an interaction involves comparing four quantities: an option "O", which scores 0 on both attributes A and B; an option "A only", which scores 100 on attribute A and 0 on attribute B; an option "B only", which scores 0 on attribute A and 100 on attribute B; and an option "AB", which scores 100 on both attributes A and B.

The magnitude of difference between option "O" and option "AB" represents the aggregate weight assigned to attribute A, attribute B, and their mutual interaction. Next, the difference between option "O" and option "A only" is assessed as a percentage of the difference from "O" to "AB"; this becomes the proportion of that aggregate weight imputed to attribute A alone. Similarly, the difference between option "O" and option "B only" is assessed as a percentage of the difference between option "O" and option "AB"; this becomes the proportion of that aggregate weight imputed to attribute B alone. Now, if the sum of the attribute A and the attribute B components is exactly equal to the aggregate weight, then there is no interaction: the incremental value of improving from a score of 0 to a score of 100 on attribute A is the same, regardless of the level of performance on attribute B, and vice versa.

If the sum of the A and B components is less than the aggregate weight, the discrepancy represents a positive

interaction component; the score of any intermediate option for this interaction component is calculated simply by multiplying that option's scores on the two component factors, and dividing the product by 100. For example, suppose it is the case that the utility to a football team of a good quarterback-receiver combination is greater than the sum of the utilities of the two individually. If the aggregate weight accounted for by the quarterback-receiver combination is 30% of the total, that means that the difference between the worst possible quarterback and receiver, and the best possible pair is worth 30 points. Suppose that the best quarterback with the worst receiver is worth 15 points, while the best possible receiver with the worst possible quarterback is worth only 5 points. Then the weight assigned to the interaction is  $30 - (15 + 5)$ , or 10 points. Now, if a mediocre quarterback (with a score of 40), and a fairly good receiver (with a score of 75) are to be evaluated as a pair, the imputed interaction component should be  $(40 \times 75) \div 100$ , or 30. Thus, of the 30 possible points which would be assigned for the best quarterback-receiver pair, this pair would receive  $(.15 \times 40) + (.05 \times 75) + (.10 \times 35)$ , or 12.75. (Naturally, this does not incorporate specific knowledge about the personal interactions between the two players involved, but only that general component of utility which reflects the complementarity of the two positions.)

A somewhat more unusual case involves attributes which to some degree duplicate one another, interfere with one another's performance, or overlap in their contributions to utility. For example, a car with good brakes and steering is less likely to need the added protection of crash-safety features than one with poor brakes and steering (or, alternatively, it would be less critical in a well-protected car to keep brakes and steering in the absolutely best possible condition). In this situation, the sum of the A-only and B-only component weights would exceed the aggregate weight

attributed to the combination of A, B, and the interaction. Therefore, the interaction term should properly receive a "negative weight." While the concept of a "negative weight" seems a bit unusual (somewhat like the mathematical concept of an "imaginary number"), there is no methodological difficulty in incorporating such a feature, and the arithmetical calculation is carried out just as in the preceding example, except of course that the interaction term will be negative. The general formulas for determining the score and weight on an interaction term are the following:

$$\begin{aligned} \text{Interaction Score} &= \text{Attribute A Score} \\ &\quad \times \text{Attribute B Score} + 100; \end{aligned}$$

$$\begin{aligned} \text{Interaction Weight} &= \text{Value of "AB"} \\ &\quad - (\text{Value of "A only"} \\ &\quad + \text{Value of "B only"}), \end{aligned}$$

where 'Value' is interpreted as the difference between the indicated option and option "O".

There seems to be no problem implementing such a feature in the current version of QVAL. The difficulty lies in the possibility that the user may misuse or overuse the feature, leading to wasted time, unnecessary complexity, and possible confusion. A reasonable suggestion would be, therefore, to perform empirical tests on an off-line version of this interaction-assessment system prior to incorporating it into QVAL or any other decision aid, and then to engage in further testing within the context of QVAL's operation. The objectives of such a testing program should be to develop criteria for when such a feature should be invoked, how best to perform the assessments, how to present preliminary results for the user's verification, and how to represent the final results in the model and in the hierarchical structure.

## 5.6 More Sophisticated Clustering Routines

Because Section 3.8 contains a fairly detailed discussion of the theory, current method, and possible alternative means of developing a sound hierarchical utility structure, this section will merely summarize the properties of a desirable clustering routine. First, it should segment the list of attributes into meaningful clusters, which can be dealt with separately from the remaining attributes. If any interactions, overlaps, duplications, etc., are expected, the interacting attributes (which are likely to be related in meaning also) should be placed together, in order to minimize the impact of any error due to ignored interactions, or to facilitate the inclusion of interaction terms as the model is further developed. Still another desirable property (closely related to the first two) is the ease of value comparison between attributes in the same category.

The information necessary to meet the three criteria--semantic relation, interaction, and commensurability--cannot be derived from the scores and weights assigned in the initial model; some additional information must be elicited from the user. Because of the time and effort involved in such elicitation, a desirable method must strive for maximum effectiveness within a limited time, and for maximum comprehensibility to the untrained user.

## 5.7 Explanation, Interpretation, and Display of Results

QVAL's current facility includes two final displays: the scores, weights, and overall values in tabular form, and the hierarchical structure with aggregate weights in outline format. While these two displays contain all of the essential information that appears in the model, additional features might do more to communicate the most important conclusions, to highlight the major factors and most likely options, and

otherwise to aid the user's comprehension and application of the results.

A first step in this direction would be a graphic display capability which could provide much more immediate visual feedback about the implications of a given model, and more compelling presentations of its results. Size, color, motion, and other visual features could be used to highlight the most important aspects of any model, and iconic images could be used to simplify and reinforce the identification of options and attributes.

Beyond the institution of graphical aids to presentation, other measures might prove valuable in helping the user to process what might still appear to be an overload of information. For example, it might be useful to abstract the critical differences which distinguish the top two options from one another (or, for that matter, any two selected options from one another). Or it might be worthwhile to summarize the performance of each option in a single paragraph of text, explaining its strengths and deficiencies in comparison with the other options.

Factor or cluster analyses of the scores (and perhaps the weights) might allow the set of options to be structured into a factorial or hierarchical organization in a way which provides some insight into the more critical underlying features which have determined the model's results. For example, suppose an analysis of the scores indicated that final overall score could be predicted almost perfectly by two "factors" (not necessarily attributes, but any properties of the options which might affect one or more attributes), as in the case of houses, where price and proximity to work might account for 90 percent of the variance on overall utility; such information might help the user in re-thinking the model, or in generating or searching for additional options which were likely to improve upon the ones studied.



Given enough sophistication, it might even be worth envisioning a system that, upon completion of an elicitation session, would produce a complete report, containing not only the results in tabular and graphic formats, but also a full text, documenting the methods used, the results and their implications, and any recommendations for continued analysis or information gathering. Naturally, developing such a system would be a major effort, predicated on a number of prerequisite technological advances, but the desirability of its goal justifies the concept's ambitiousness.

#### 5.8 Artificial Intelligence Programs to Aid Decision Modeling

The current development of new techniques in artificial intelligence will eventually provide a number of highly desirable capabilities to users of programs such as QVAL. New programs will be able to process and to produce messages in a much less constrained natural language (eventually communicating by audible speech, rather than the printed word). They will have sophisticated pattern-recognition and feature-analysis capabilities for analyzing complex data. They will allow for reference to visual displays, using either physical cues (such as pointing to or touching an item) or verbal cues (such as referring to it in speech).

The artificial intelligence programs of the future will probably also be able to capture and duplicate (and perhaps even to improve upon) the heuristic strategies employed by the trained decision analyst, providing a much more sophisticated approach to the entire problem. Furthermore, by performing tentative analyses on a continual basis, they will be able to determine which topics are likely to benefit most from additional analysis or information gathering, and which should be left as they stand.

Finally, future programs will have the ability to access large and complex data bases, including a wide variety of facts about substantive application areas, about the semantics of the language and special vocabulary of the user, and about their own prior experience in performing decision analysis. By continually learning, requesting information and updating their knowledge base, such computer programs will eventually become not only competent decision analysts, but also experts in the substantive areas of interest to the users. Naturally, even the most sophisticated and intelligent computer routines would still depend on the user's personal values and judgments of the specific facts of the problem situation, but much of the initial period during which the human analyst must familiarize himself with the problem area could be avoided, and much data-gathering could be eliminated.

Although some of the ideas discussed in this section may strike the reader as impractical or visionary, most of the foundation work in machine language, perception, and problem solving has begun already. As time progresses, the machines themselves, the programs available, and the knowledge of the artificial intelligence community will increase in sophistication until eventually a synthesis of the right components does realize the goals outlined here.

#### 5.9 Accommodating Imprecision in Assessments

Although the numerical assessments which form the basis for most decision analyses require the specification of a single value for each score or weight required, the numbers elicited cannot justifiably be said to represent the exact values they are supposed to. Apart from the noise and the biases which are inherent in any empirical measurement (and particularly in subjective judgments), a variety of other sources contribute to imprecision. The capacity of the

human brain to distinguish between levels of value (or even to distinguish between levels of directly observable phenomena such as the loudness of a noise) is limited to a certain resolution. The context of an evaluation may not be perfectly well defined; and depending on differences of interpretation or actual fluctuations in circumstances, the scores and weights in any model may have to be changed in an unpredictable manner. Furthermore, uncertainties about future events which might effect the valuation of the options (if not explicitly modeled using probability theory) may induce still greater variability on the "true" values measured.

One method for dealing with this variability is to accept the elicited values as provisional, and to use sensitivity analysis after the fact to test the robustness of the results as scores and weights are varied. A thorough sensitivity analysis, however, would incorporate a great deal of additional effort, requiring a number of highly difficult and time-consuming elicitations, as well as the introduction of continuous probability distributions into a problem where they are not of central importance. In particular, a thorough sensitivity analysis would have to assess the joint probability distribution of all scores and weights (plus, perhaps, errors, biases, and overlooked attributes), to determine the overall probability that the option with the highest overall score would still remain best, and to assess the potential regrets or adverse consequences which could arise from adopting the indicated option when in fact the true situation favored a different one.

If all of the measurements involved in a numerical decision analysis were subject only to random, independent errors, it would be quite reasonable to assess the probability distribution for each of the variables and multiply them together to obtain the joint distribution. However, in most cases, there are systematic methodological biases which

affect a number of variables in the same way, and substantive facts and value judgments which impact upon more than one option or attribute. If these dependencies could be identified and measured, again a probabilistic sensitivity analysis might be quite practical. However, the difficulty of measuring such factors, together with the problem of explaining them to a naive user (particularly without a trained analyst present), renders this strategy unusable as a general approach.

A different approach to modeling imprecision in numerical models utilizes the Theory of Fuzzy Sets, as initially formulated by L. A. Zadeh (1965). By specifying a function called the possibility distribution for each input, rather than a precise number, the user can communicate not only his "best guess" at the proper value, but also the range of his uncertainty. While the mathematical operations underlying "Fuzzy Decision Analysis" are beyond the scope of the present report (for a full description, see Watson, et al., 1979), there is an established method for representing and manipulating membership functions, resulting in an analysis which tells not only the "best guess" at the proper option, but also the degree to which the indicated variability on the inputs might affect that choice.

Fuzzy Set Theory and its applications are still in an early phase of development (only fourteen years have passed since the initial concept), but research on its mathematical foundations, its application to decision and control problems, and its psychological foundations is expanding rapidly. Although the theory is still usable only on an experimental basis, and although only preliminary steps have been taken towards developing a standard procedure for defining and measuring possibility distributions (Sticha, et al., 1979), the context of QVAL may represent an excellent testing ground for this newly developing theory, simultaneously

benefitting both the art of decision aiding and the Theory of Fuzzy Sets.

One additional benefit of the Theory of Fuzzy Sets is its ability to incorporate verbal responses to quantitative questions, in a manner which allows consistency and accuracy without demanding unrealistic degrees of precision. The ability to encode scores and weights as words or phrases, and to allow the user to communicate by using these verbal responses, would significantly enhance the attractiveness of a decision aid to the user with little quantitative training, who might feel uncomfortable being "pinned down" to numerical assessments that appear more precise than his knowledge justifies.

As the art of decision aiding and the Theory of Fuzzy Sets develop side by side, it will be useful to consider their potential for constructive interaction. Possibly, the ability to utilize the computer's computational speed to perform the mathematical operations involved in "Fuzzy Decision Analysis," combined with the user-oriented outward simplicity of approximate verbal responses, might provide the most effective approach to decision analysis for the naive user.

## 6.0 GENTREE: A CONTENT-ORIENTED AID TO DECISION PROBLEM STRUCTURING

### 6.1 Introduction

Experienced decision analysts often observe that, despite the claim that decision analysis can be applied with equal effect to problems in any area, the most successful analyses involve substantive areas with which the decision analyst has some prior experience or personal expertise. Apart from merely saving familiarization time (learning terminology, basic facts, and so forth), experience gives the analyst an intuitive grasp of the problem which prompts more perceptive questions, better suggestions, and the general ability to determine how well the current model is capturing the essence of the client's problem.

Von Winterfeldt (1979) has pointed out a contrast between two philosophies of decision analysis: an "elicitation philosophy" which selects structural frameworks first, and then uses the expertise of the decision maker and his staff to determine the numerical values required; and a "prototypical structure" approach, which uses substantive knowledge about the problem area to categorize the situation with respect to a pre-existing "tool kit" of prototypical models (e.g., facility siting, contingency planning, budget allocation, and regulation). His thesis that the latter approach tends to yield more appropriate models is a difficult one to test empirically, because in practice, increased decision-analytic sophistication and increased knowledge of typical substantive areas tend to develop side by side; the dichotomy between methodologists and substantive experts is not a realistic one.

In the context of machine-aided decision structuring, however, it is possible to consider the possibility of two

extreme approaches: the first (typified by QVAL, MAUD, and a number of other decision aiding devices) operates in complete ignorance of the semantic and substantive framework of the problem, assuming that the user's expertise and effort will incorporate any necessary information into the elicitation; the second (typified by DDI's SURVAV aid [Barclay and Randall, 1977], which is specifically designed to route ships so as to avoid satellite surveillance) incorporates a highly specific model of the problem, so that a minimum of judgment need be exercised at the time of operation. To the degree that the problem can be specified in detail before an actual decision is needed, the latter approach works well; however, most decision problems have unique properties which, despite their partial similarity to previously analyzed problems, render them sufficiently different that no pre-canned aid would be satisfactory.

The technology of decision aiding thus faces a dilemma: if a model is constructed in such a way as to capture essential substantive knowledge about the problem area, its application is extremely limited, and very little transfer of experience to related problems is possible; on the other hand, if a general-purpose model is developed in the absence of any specific substantive content, the model is likely to remain insensitive to the specific nuances of the problem, relying heavily on the ability of the user to incorporate all of his expertise into a structure that may not match the way he thinks.

This dilemma is analogous to one encountered in the field of automatic language comprehension: syntactically oriented, content-free language processors tend to be slow, inefficient, and error-prone, while semantically oriented systems which depend heavily upon context tend to work well only when limited to a rather narrow range of situations. As an extreme case, a "language comprehension" program which

simply contained a list of the allowable messages and their proper responses would work quickly and accurately, as long as acceptable messages were received; on the other hand, even with an extremely large memory, the most rapid machines would ultimately be limited in retrieval time as the list of allowable messages was increased. A more general routine, operating from general syntactic principles and utilizing semantic knowledge only after "parsing" its input, might operate extremely slowly, due to the necessity to reconcile ambiguities, to investigate possible interpretations that turn out to be incorrect, etc.

In order to utilize both the specific semantic details of a sentence's context and the syntactic information of its structure in an efficient way, Roger Schank (1972, 1973) has proposed a variety of language comprehension schemes based upon two important concepts: (1) information about the substantive world can be represented in a structured format which describes the large number of highly specific, concrete terms (e.g., specific objects or people) in terms of a relatively small number of "conceptual relationships" (e.g., time sequence, location, etc.); and (2) sentences can be interpreted in terms of their implications within this conceptual scheme of representation. Thus, as a sentence is processed, the conceptual relations already known to hold with respect to the elements of the sentence lead to an active search for missing data which is likely to be needed.

At a higher level still, the concept of organization into "scripts" provides a generic framework for certain somewhat stereotyped situations, in which a few basic relationships, roles, or objectives are nearly universal, despite wide variation in details. For example, a "restaurant" script might involve the following elements: a customer or group of customers, generally motivated by hunger; a waiter or waitress whose job is to serve them food and



beverages, for which the server is paid both wages and tips; an owner receives the customer's money, and pays for the food and the employees' wages; and a cook, who prepares the food for the customer, at the request of the waiter or waitress, in exchange for a salary from the owner. This complex set of relationships (and the implied connections with other scripts such as "employer-employee relationships" or "cooking food") can be represented structurally, in a manner which is midway between the purely syntactic, content-free system, and the purely semantic, situation-specific information retrieval system.

Returning to decision analysis, we observe the following trade-off: as more complete and specific information about previous analyses is stored and retrieved, the requirement for newly elicited information decreases; but at the same time, the effort (and the associated time and expense) of organizing and retrieving that information increases, and the usefulness (generalizability) of the additional information decreases. Thus, at one extreme, there is the decision aid which takes no account of past experience or prior knowledge, and therefore deals with each problem as if it were a technically proficient but substantively naive novice. At the opposite extreme would be a program which could directly store the structure, scores, weights, and rationale for every decision analysis it had performed, and access that information through an interactive lookup system (a number of such schemes might be possible); that system would involve a great deal of storage and retrieval effort, and even when an existing structure was sufficiently related to the current problem to indicate its retrieval, there might be no obvious way to distinguish that part of its information which is relevant to the problem at hand from that which is not, or to apply that information to the current analysis.

A reasonable compromise would be to store selected information about decision problems, at a level of generality which permits easier application of the information retrieved. In terms of efficiency, it is often the case that a very small percentage of the total information available on a given topic can account for nearly all of the ability to discriminate among options, so that a suitably chosen storage and retrieval strategy could optimize the speed and usefulness of a decision-aiding system.

This, then, is the motivation for GenTree (Generic Tree-Structuring Aid). Currently, GenTree exists only as a conceptual design; however, there seems to be no technical obstacle to its realization within the bounds of existing technology. Any reference to more sophisticated techniques such as artificial intelligence, interactive graphics, or video disc technology in the discussion to follow are purely for illustrative purposes, and in no way restrict the concept's realization to dependence on such techniques.

## 6.2 Theoretical Analysis

The critical feature which distinguishes GenTree from QVAL, MAUD, and other decision aids is GenTree's ability to extract substantive information from its "knowledge base" in order to contribute more effectively to the construction of an appropriate hierarchical utility model. GenTree must, therefore, possess the following three capabilities:

1. it must be able to encode and store information about previously completed analyses in an accessible manner;
2. it must be able to determine, based on current knowledge about the ongoing analysis, what items in its knowledge base are relevant, and how they relate to the current problem; and

3. it must be able to utilize the retrieved information in a way which facilitates the construction of a useful and valid model.

In order to perform the functions of encoding, retrieval, and application in a timely and economical manner, GenTree must rely on an overall structure or framework for representing models, their elements, and their descriptive features in a generic way. Human thought does not operate by remembering every fact about every object; economy dictates that we categorize objects into a relatively small number of generic classes, assign names to those classes, and then construct sentences from the limited vocabulary by applying a fairly small, highly structured set of rules (e.g., transformational grammars). Applying a similar strategy to the more restricted domain of utility models, GenTree will have to categorize the fundamental elements of a problem into generic units, represent those units in a symbolic form, and use a well-structured set of rules to organize, combine, and manipulate them.

GenTree deals with its task by incorporating two kinds of information into systems: specific information about elements which have appeared in prior models will be represented in the context of a Data Representation System (DRS), while more macroscopic information describing entire models and their contexts will be organized into a Semantic Context Framework (SCF). By orienting an ongoing analysis in terms of the existing Semantic Context Framework, GenTree can restrict its attention to that portion of the DRS which is most likely to be applicable to the current problem.

The remainder of this discussion will focus on three topics: how a Data Representation System may be organized to provide efficient access to stored information; how a Semantic Context Framework can be structured to facilitate the search for relevant information within the DRS; and how

GenTree as a whole might operate in order to utilize both knowledge bases effectively.

6.2.1 Data representation system - A number of possible configurations could be envisioned for a DRS to incorporate the information accumulated during past analyses. One highly promising method is based on the notion of a Case Grammar, as described by Fillmore (1968), and elaborated in Schank's Conceptual Dependency models (Schank, 1972). The fundamental principle behind such an approach is that although the topics of reference may vary widely, the number of generic relationships among the elements of any utility analysis can be relatively small, perhaps no more than one or two dozen.

In terms of a hierarchical utility model, the element, or unit of analysis, may be considered as a node on a tree. Structurally, the fundamental relationship from which a tree can be defined is that between a superordinate ("parent") node, and an immediately subordinate ("child") node. Because the structural "parent-child" relationship itself contains no substantive information, it is necessary to add functional relationships which explicate the underlying substantive content. Thus, the basis for GenTree's DRS is a set of conceptual, functional, semantic operations which correspond to the "deep structure" behind the superficially simple structural, syntactic parent-child relation.

Perhaps the simplest conceptual relationship that appears in hierarchical utility structures is the partition of a set or category into subsets or subcategories. For example, a node labeled "United States" may be subdivided into nodes labeled "Northeast," "Southeast," "Central," "Northwest," "Southwest," "Alaska," and "Hawaii". The first five of these represent aggregations which may in turn be subdivided into their component states, and if necessary, each state could be subdivided into countries or other smaller units.

Now, suppose that on one analysis, it was noted that the United States was subdivided into the six regions specified. It would be useful to recall that subdivision, including the names of the regions, any time the node "United States" appears in future models, to display this as a possibility, and to allow the user to decide whether or not to incorporate it (either intact or after modifications). It would also be useful to store the more general observation that geographic regions (a generic category into which the United States falls) have in the past been divided into smaller subregions (such as the six specified ones), with no reference to the particular region being analyzed.

Another similar relationship might divide a system into subsystems (e.g., an automobile into its electrical, fuel, exhaust, etc., systems), a period of time into smaller units, a population into subcategories, and so forth. Any time the utility of a given item can be represented as the sum of the utilities of its components (appropriately weighted), this set-subset decomposition is indicated.

While the set-subset relation represents a decomposition of the item being evaluated, other generic types of decomposition can apply. An option may be evaluated according to the various groups or individuals it affects (users, or constituencies). Its utility may be subdivided according to the tasks it performs, as in a mission-area analysis. It may be evaluated in terms of a number of goals, or over a number of time periods. Despite the multitude of specific subdivisions that have been used in models, nearly all of them fall within a small number of generic categories such as the ones discussed here.

The context of a given analysis may indicate which modes of decomposition are likely to be useful, based on past experience. Thus, in evaluating automobiles, if a

previous analysis found it useful to consider each subsystem separately, the same strategy might work well in a current analysis; a tentative list based on the prior model could be of use as a starting point, even if some modifications had to be made. Furthermore, even if the current problem involves evaluating vehicles of a different type (e.g., boats), the subsystems may differ while the overall strategy and the list of automobile subsystems might be used as an analogy to prompt the generation of a comparable list for boat subsystems.

After such an exercise, it might be valuable to abstract the areas of commonality, by indicating, for example, that both cars and boats (at least, power boats) have fuel systems, propulsion systems, steering and speed controls, passenger compartments, baggage compartments, etc., and that these simply have different names dependent upon the context. As more information is acquired, greater generality can be developed, with less dependence on context to find an exact match with the stored information. (How this generality could be encoded and utilized depends on the Semantic Context Framework, discussed in Section 6.2.2).

It is possible to represent the information contained in a hierarchical utility structure as a list of "decomposition schemata," each of which specifies the following information:

- o the identifying name of the parent node;
- o the context (a summary of the parent node's "lineage," and its function with respect to higher-level nodes);
- o the mode of decomposition employed; and
- o the identifying names of the "children" nodes.

Thus, a possible entry might appear as follows:

FUEL SYSTEM

Context: Subsystem of MOTOR VEHICLE

Mode of Decomposition: System into subsystems

Decomposition elements: FUEL TANK, FUEL LINE,  
FUEL PUMP, CARBURETOR.

Another example, illustrating a different kind of decomposition, might be the following:

COMPUTER SYSTEM

Context: System to be evaluated

Mode of Decomposition: System by constituencies

Decomposition elements: PURCHASING COMPANY,  
SYSTEMS PROGRAMMERS, APPLICATION PROGRAMMERS,  
CUSTOMERS.

Having represented information in this form, GenTree can retrieve that knowledge the next time a similar context arises, thus providing the user with at least an initial suggestion about a possible continuation for his analysis.

6.2.2 Semantic context framework - In order to make most effective use of the information stored in the Data Representation System, GenTree should have a second kind of knowledge base, incorporating "semantic" information about the models and contexts that have been evaluated. As Collins and Quillian (1969) have indicated, it may be extremely economical to collect information about a generic class of items, and to store and access that information via the generic class, rather than at each individual item's location. Thus, the information that a robin has wings could properly be inferred from the fact that birds in general have wings, and the "semantic relation" which designates "robin" as a subclass of the set of birds.

The basic relationship to be represented in the SCF is the set-subset relation. When evaluating objects of a given generic type, GenTree can search not only for occurrences of the specific objects (an unlikely event, unless the range of substantive applications is quite narrow), but also for the smallest immediately superordinate category, and then for all categories superordinate to that one.

Once a problem has been formulated for analysis, GenTree will attempt to locate it within a generic framework, by asking questions which will establish linkages to existing categories. For example, GenTree might print out a list of "keywords" and ask the user to indicate which if any of those keywords apply to the current problem. As an alternative, GenTree might ask a series of taxonomic questions (this would correspond to a "top-down" attempt at categorization, as opposed to the "bottom-up" method of starting with keywords). In either case, GenTree would establish, wherever possible, the most specific superordinate category that applies to the options being evaluated.

The categorization process applies not only at the top level, but at all stages of the model analysis: as a node is subdivided, its components may also be located in the "semantic space" defined by the SCF. For example, when the Name "DOWN PAYMENT" appears in the evaluation, GenTree should be able to ascertain that a superordinate category is likely to be a "MONETARY COSTS/IMMEDIATE," which in turn would be included in "MONETARY COSTS." This could prompt a tentative listing of other types of cost breakdowns, by time period, type of cost, or source of funding.

Of course, when GenTree does find an exact match or synonym for the node under discussion, it can provide immediate access to superordinate categories and to possible



subdivisions. Sometimes, even in the absence of such a convenient circumstance, it is possible to "attach" a tentative connection between the node currently under scrutiny and some other analogous node already in the SCF (e.g., by matching patterns of keywords). Then, a direct query to the user might serve to establish a new node in the SCF system, thereby permitting access to the relevant superordinate category. For example, if the items to be evaluated are trucks, and no prior reference to the node "TRUCK" exists, it might nonetheless be possible, through keyword matching or top-down questioning, to determine an essential similarity to the already existing node, "AUTOMOBILES"; then, adjustments could be made to indicate that both "TRUCKS" and "AUTOMOBILES" fall into a new superordinate category ("MOTOR VEHICLES"), with their common features stored at this superordinate level, and their distinguishing characteristics stored at the lower-level nodes.

By a continuing process of checking the SCF for relevant superordinate categories, searching the DRS for potentially useful information about the categories detected, interactively determining how (if at all) to use that information, and storing any newly obtained information for future use, GenTree will make efficient use of the data collected previously, as it accumulates new information for future applications. Depending on the extent and complexity of the applications it encounters, GenTree may eventually build up a knowledge base comparable to that of the most experienced decision analysts. Of course, by that point there may arise an information capacity problem; but as computers grow faster, larger, and cheaper, and as information processing technology develops more sophistication, the capacity problem should diminish in importance.

6.2.3 Overall mode of operation - In the discussions of the DRS and the SCF, we have alluded in general to the

procedures that might be built into GenTree. While the concepts have not yet been developed to the point where a single "best" approach can be identified, this section describes a reasonable first step in that direction.

At the outset of a new problem, GenTree will ask the user to identify the major options to be evaluated, and then try to elicit a generic name for that set of options. If no immediate generic name can be found to match an entry in the SCF, GenTree will proceed through a top-down taxonomic approach, asking questions like the following:

Please select the word or phrase that best completes the following sentence:

Each of the options being considered is . . .

- (1) A weapons system.
- (2) An information processing system.
- (3) A transportation vehicle or system.
- (4) A course of action.
- (5) Other.

If it is possible to narrow down the description to a more specific category, GenTree can then attempt further specification with additional questions.

When the limit of specificity has been reached under the top-down approach, GenTree may then present a "shopping list" of keywords which might relate to the options under consideration. To the degree that the taxonomic questions have narrowed the focus, the keyword list might be "filtered" somewhat to reflect the range of interest indicated. The net result of this procedure will be (unless the decision involves a completely novel set of items) an identification of the current problem in terms of the existing SCF.

The next step in the procedure is to determine in what ways the top-level node might be disaggregated. Referring to the DRS, GenTree may observe, for example, that

a weapons system is typically analyzed into sub-systems; evaluated according to the situations in which it will be applied; or treated in terms of its goals or missions. Furthermore, if the weapons system is more specifically defined (e.g., if only surface-to-surface missiles are being considered), one or more past lists of subordinate nodes may be retrieved as suggestions for the current analysis.

In any event, GenTree will interactively attempt to establish which, if any, of the indicated modes of sub-categorization is applicable to the current problem. If more than one type of classification is possible, GenTree will also assess whether the order of subdivision is arbitrary, or whether there is a natural way of subordinating one mode of decomposition to another. As the list of sub-components is elicited, the names thus derived are (tentatively) entered into the DRS, and also looked up on the SCF. If a match is found, suggestions for further subdivision may be presented to the user, so long as the context specified in the model agrees with that stored in the DRS.

Ultimately, it would be desirable for GenTree to be able to use guidelines about the time and resources available to help limit the model to a reasonable level of detail. For the present, however, it should suffice to ask the user if he is satisfied with the current level of detail, and if so, to begin to assess numerical values, while keeping open the option of a more detailed continuation of the modeling at a later time. (A still more sophisticated approach might determine the criticality of each node, and continue to disaggregate only those nodes that seem to be worthwhile, in a value-of-information sense.)

Currently, GenTree's performance will stop when a structure has been defined and approved by the user. At that point, any newly assessed information that could be

used to augment the SCF and the DRS will be saved, and the current model presented in a format corresponding to the output of a standard structuring routine, such as the "BUILD" portion of DDI's EVAL program (Allardyce et al., 1979).

6.2.4 Alternatives and extensions - The procedures outlined in the preceding sections represent a more or less naive approach to decision problem structuring aids, in the sense that most of the procedures used are fairly straightforward template-matching or feature-counting methods of pattern recognition, and most of the machine-user interaction is highly restricted in structure and content. While these methods may suffice to demonstrate the feasibility of the GenTree approach, it is clear that current techniques in Artificial Intelligence, particularly in the fields of pattern recognition and machine language and dialogue, could greatly enhance GenTree's usefulness, capacity for enriched interaction, and general range of applications. In particular, the work of Winograd (1970, 1972) on machine language, that of Schank and his associates on scripts and conceptual dependency (Schank, 1972, 1973), and that of Simon and Newell (1971) on heuristic strategies for problem-solving, should all be considered for their potential impact on the success of a GenTree-like approach to interactive problem structuring.

Even within the limited context of GenTree's present definition, substantial room for human-engineering studies exists. Complex trade-offs must be made involving training time, speed of operation, machine capacity, simplicity of operation, and range of applications, to name a few criteria. Only direct empirical observation will provide the information necessary to design an effective system.

A final area which holds a great deal of promise in any attempt to implement a GenTree-like system is the field of advanced interactive computer graphics. Techniques

such as the Spatial Data Management System (SDMS), (Negroponte, 1979), could allow GenTree to break free of the linear flow of thought (and the linear mode of search) currently designed into GenTree, thus allowing the system to take fuller advantage of the user's own creativity and knowledge. Furthermore, by its greater flexibility of display and input elicitation, an interactive graphics system would allow substantially easier communication with the user, and limit the purely mechanical problems of the dialogue to a minimum.

### 6.3 A Notional Example

This section presents a concrete example of how a GenTree system might operate. No claim is made that the procedures illustrated are necessarily the best ones, nor is anything implied about the time, computer memory, etc., needed to carry out their implementation. Rather, the purpose of this section is to provide a basis for future discussion and planning, and a "straw man" to focus the comments, suggestions, and criticisms that may arise.

GenTree, as illustrated here, will consist of three components: the Data Representation System (DRS), as described in Section 6.2.1; the Semantic Context Framework (SCF), as described in Section 6.2.2; and a Current Model Representation (CMR) which assembles information components as provided by the user or retrieved from the DRS and the SCF, to form a hierarchical utility model of the current problem. The immediate end-product will be the fully developed utility model for the user's problem; however, a dynamic version of GenTree will also automatically update the SCF and DRS to incorporate the newly elicited knowledge, thereby providing a valuable by-product.

Suppose that the problem at hand involved selecting an electric vehicle for moving supplies and heavy objects

around a military base. GenTree might begin with a list of very general categories, and determine that the objects to be evaluated are all instances of the category SYSTEM. Searching the SCF, GenTree might ask for further specification of the kind of system (TRANSPORTATION SYSTEM, as opposed to WEAPONS SYSTEM or INFORMATION PROCESSING SYSTEM). Under the SCF entry for TRANSPORTATION SYSTEM, GenTree might further interact with the user to determine that the items were instances of MOTOR VEHICLE, and in particular, of WHEELED LAND MOTOR VEHICLE.

At this point, the CMR contains the following representation:

X (NEW MODEL)

Superordinate categories: WHEELED LAND MOTOR  
VEHICLE, MOTOR VEHICLE, TRANSPORTATION SYSTEM,  
SYSTEM.

Knowledge:

Missions: transport people; transport objects.

Functions: load; start; transit; park; unload.

Subsystems: original energy source; engine  
(transforms original energy into mechanical  
energy); drive train (translates mechanical  
energy into locomotion); controls; passenger  
seats/compartment; cargo/luggage compartment;  
safety features; comfort features.

Constituents: owner; operator(s); passengers;  
cargo beneficiaries; maintenance personnel.

Cost/Benefit: monetary; non-monetary.

So far, only the categorical information from the CMR has been utilized. The next step will be to establish a top-level node called OVERALL UTILITY, and to present the various knowledge items to the user, for three purposes: verification; specification of components and technical terms; and determination of relevance to the immediate decision problem. For example, the following dialogue might take place:

G (GenTree): Please supply a name for the items being evaluated, in plural and singular forms. (Plural)  
All items are ...

U (User): ELECTRIC TRUCKS

G: (Singular) Each one is ...

U: AN ELECTRIC TRUCK

G: As a transportation system, an electric truck might have the following missions: (1) transport people; (2) transport objects. Are these missions both relevant to OVERALL UTILITY?

U: NO

G: Please type the number(s) of any missions which are relevant (if none, type 0).

U: 2

G: Apart from "transport objects" are there any other missions which might be relevant to OVERALL UTILITY?

U: NO

G: In previous analyses, missions have sometimes been subdivided into categories according to the following schemes:

1. time of day
2. time of year
3. weather/environment
4. geographic location
5. criticality
6. sub-tasks

Please type the number(s) of any of these schemes which might be worth considering as an expansion of "transport objects" (if none, type 0).

U: 6

G: Please list the component sub-tasks into which "transport objects" might be divided.

U: TRANSPORT SUPPLIES  
TRANSPORT HEAVY OBJECTS

G: The mission "transport objects" has been subdivided into "transport supplies" and "transport heavy objects." Do you wish to continue subdividing either of these two?

U: NO

G: As a transportation system, an electric truck's operation might be subdivided into the following functions: (1) load; (2) start; (3) transit; (4) park; (5) unload. Are these functions all relevant to OVERALL UTILITY?

U: YES

G: Are there any other functions you would like to add to the list?

U: NO

G: Do you wish to divide any of the functions into component sub-functions?

U: NO

G: As a wheeled motor vehicle, an electric truck might be divided into the following sub-systems:

- (1) original energy source
- (2) engine (original energy into mechanical)
- (3) drive train (mechanical energy into locomotion)
- (4) controls
- (5) passenger seats/compartment
- (6) cargo/luggage compartment
- (7) safety features
- (8) comfort features

Would a division into these or some other similar set of subsystems be of value in this decision?

U: YES

G: Are all of the named subsystems relevant to the assessment of OVERALL UTILITY?

U: NO

G: Please type the number(s) of those subsystems which might be worth evaluating.

U: 1 2 3 4 6 7

G: Are there any other subsystems which should be added to the ones just specified?

U: YES

G: Please list the additional subsystems.

U: AUTOMATIC UNLOADER



G: The current list of subsystems is:

- (1) original energy source
- (2) engine
- (3) drive train
- (4) controls
- (5) cargo/luggage compartment
- (6) safety features
- (7) automatic unloader

Do you wish to divide any of these systems into sub-components?

U: YES

G: Please type the numbers of those you do wish to subdivide.

U: 1 4 6

G: Please list the sub-components of "original energy source."

U: BATTERY  
RECHARGER

G: Please list the sub-components of "controls."

U: STEERING  
BRAKES  
ACCELERATOR

G: Please list the sub-components of "safety features."

U: SIGNAL/WARNING LIGHTS  
EMERGENCY BRAKES  
DRIVER PROTECTION IN COLLISIONS

G: As a vehicle, an electric truck may have utility with respect to one or more of the following constituents:

- (1) owner
- (2) operator(s)
- (3) passengers
- (4) cargo beneficiaries
- (5) maintenance personnel

Do you wish to subdivide OVERALL UTILITY according to the constituents on this list, or a similar list?

U: NO

G: As a system, an electric truck's OVERALL UTILITY may be subdivided into (1) monetary costs/benefits, and (2) non-monetary costs/benefits. Do you wish to make this distinction?

U: YES

G: The MONETARY COST/BENEFIT of a system is often subdivided into components. The following modes of analysis may be useful: (1) time periods; (2) funding cycle; (3) constituents; (4) subsystems. Please type the number(s) of any of these that might be worth considering.

U: 2

G: As a motor vehicle, an electric truck's MONETARY COST may be subdivided into the following funding stages: (1) Research, Development, Testing, and Evaluation; (2) Procurement; (3) Operation and Maintenance. Please type the numbers of the stages which are relevant to the current decision.

U: 2 3

G: Are there additional phases you would like to add?

U: NO

G: Would you like to subdivide Procurement into components?

U: NO

G: Would you like to subdivide Operation and Maintenance into components?

U: YES

G: As a wheeled land motor vehicle, an electric truck may have O&M COSTS/BENEFITS such as the following: (1) operating personnel; (2) repairs; (3) routine maintenance; (4) fuel; (5) replacement tires or treads; (6) spare parts; (7) training and instruction personnel. Please type the number(s) of the components that are relevant to the current analysis.

U: 2 3 4 5 6

G: Are there any other components of O&M COST/BENEFIT to be considered?

U: NO

G: As a system, an electric truck has a NON-MONETARY UTILITY which may be subdivided in the following ways: (1) by missions; (2) by functions; (3) by subsystems; (4) by constituents; (5) by time

periods. Please type the numbers of the modes of analysis which might be relevant to the assessment of an electric truck's NON-MONETARY UTILITY.

U: 1 2 3 4

G: According to previous responses, NON-MONETARY UTILITY may reflect as many as

2 missions  
5 functions  
12 subsystems and  
4 constituents.

If all possible combinations are assessed, this will divide NON-MONETARY UTILITY into 480 parts.

The number of components for NON-MONETARY UTILITY may be reduced in three ways:

- (1) grouping the items in one of the lists (e.g, the subsystem list) into fewer categories;
- (2) identifying situations in which two or more modes of analysis are completely independent of one another and may therefore be done in parallel; or
- (3) "pruning" the structure by eliminating combinations which are impossible, irrelevant, or meaningless.

Would you like to try any of these methods for shortening the list of sub-categories?

U: YES

G: To start, please consider the list of missions:  
(1) transport supplies; (2) transport heavy objects. Do you wish to combine these two missions into a single category?

U: NO

G: Next, please consider the list of functions.  
There are currently five functions: (1) load; (2) start; (3) transit; (4) park; (5) unload.  
Do you wish to combine any of these functions into a larger category?

U: YES

G: Please type the numbers of the functions to be combined.

U: 1 5

G: Is LOAD/UNLOAD a satisfactory name for the new category of functions?

U: YES

G: Do you wish to combine any other groups of functions into a larger category?

U: NO

G: The current list of functions now appears as follows: (1) start; (2) transit; (3) park; (4) load/unload. Do the relative importances of these functions change significantly depending on the mission being considered?

U: NO

G: Next, please consider the list of subsystems. There are currently 12 subsystems, some of which are parts of larger categories:

- (1) original energy source
  - (1.1, battery);
  - (1.2, recharger);
- (2) engine;
- (3) drive train;
- (4) controls
  - (4.1, steering);
  - (4.2, brakes);
  - (4.3, accelerator);
- (5) cargo/luggage compartment;
- (6) safety features
  - (6.1, signal/warning lights);
  - (6.2, emergency brakes);
  - (6.3, driver protection in collision);
- (7) automatic unloader.

Do you wish to recombine the components of any of the compound subsystems in order to reduce the size of the list?

U: YES

G: Do you wish to recombine the components of (1) original energy source?

U: NO

G: Do you wish to recombine the components of (4) controls?

U: NO

G: Do you wish to recombine the components of  
(6) safety features?

U: YES

G: The current list of subsystems is: (1) battery;  
(2) recharger; (3) engine; (4) steering;  
(5) brakes; (6) accelerator; (7) cargo/luggage  
compartment; (8) safety features; (9) automatic  
unloader.

In evaluating NON-MONETARY UTILITY, would it be  
simpler to trade off "battery" or "recharger"  
individually versus the other items, than to  
compare "original energy source" to the other  
items?

U: YES

G: In evaluating NON-MONETARY UTILITY, would it be  
simpler to trade off "steering," "brakes," and  
"accelerator" individually versus the other items,  
than to compare "controls" to the other items?

U: YES

G: Do the relative importances of the subsystems  
change significantly depending on which of the  
two missions is being considered?

U: YES

G: Do the relative importances of the subsystems  
change depending on which function is being  
considered?

U: YES

G: Are there any missions, functions, or mission-  
function combinations for which some of the  
subsystems may make no contribution at all?

U: YES

G: Please type the number(s) of any subsystems  
which do not affect NON-MONETARY UTILITY,  
restricted to the mission "transport supplies."

U: 0

G: Please type the number(s) of any subsystems  
which do not affect NON-MONETARY UTILITY,  
restricted to the mission "transport heavy  
objects."

U: 0

G: Please type the number(s) of any subsystems which do not affect NON-MONETARY UTILITY, restricted to the function "load/unload."

U: 1 2 3 4 5 6

G: Please type the number(s) of any subsystems which do not affect NON-MONETARY UTILITY, restricted to the function "start."

U: 3 4 5 6 7 8 9

G: Please type the number(s) of any subsystems which do not affect NON-MONETARY UTILITY, restricted to the function "transit."

U: 1 2 9

G: Please type the number(s) of any subsystems which do not affect NON-MONETARY UTILITY, restricted to the function "park."

U: 1 2 3 5 6 7 9

G: In addition to any missions or functions which serve to eliminate a subsystem from consideration, are there any specific combinations of a mission and a function for which one or more subsystems would have no effect on NON-MONETARY UTILITY?

U: NO

At this point in the analysis, GenTree has sufficient information to propose the structure displayed in Figure 6-1. Although the present illustration keeps the original order of analysis (cost/benefit, mission, function, subsystem) intact, more sophisticated routines could be incorporated that could determine which of several equivalent presentations would "optimally" summarize the evaluation (perhaps by displaying all possible arrangements, and letting the user choose).

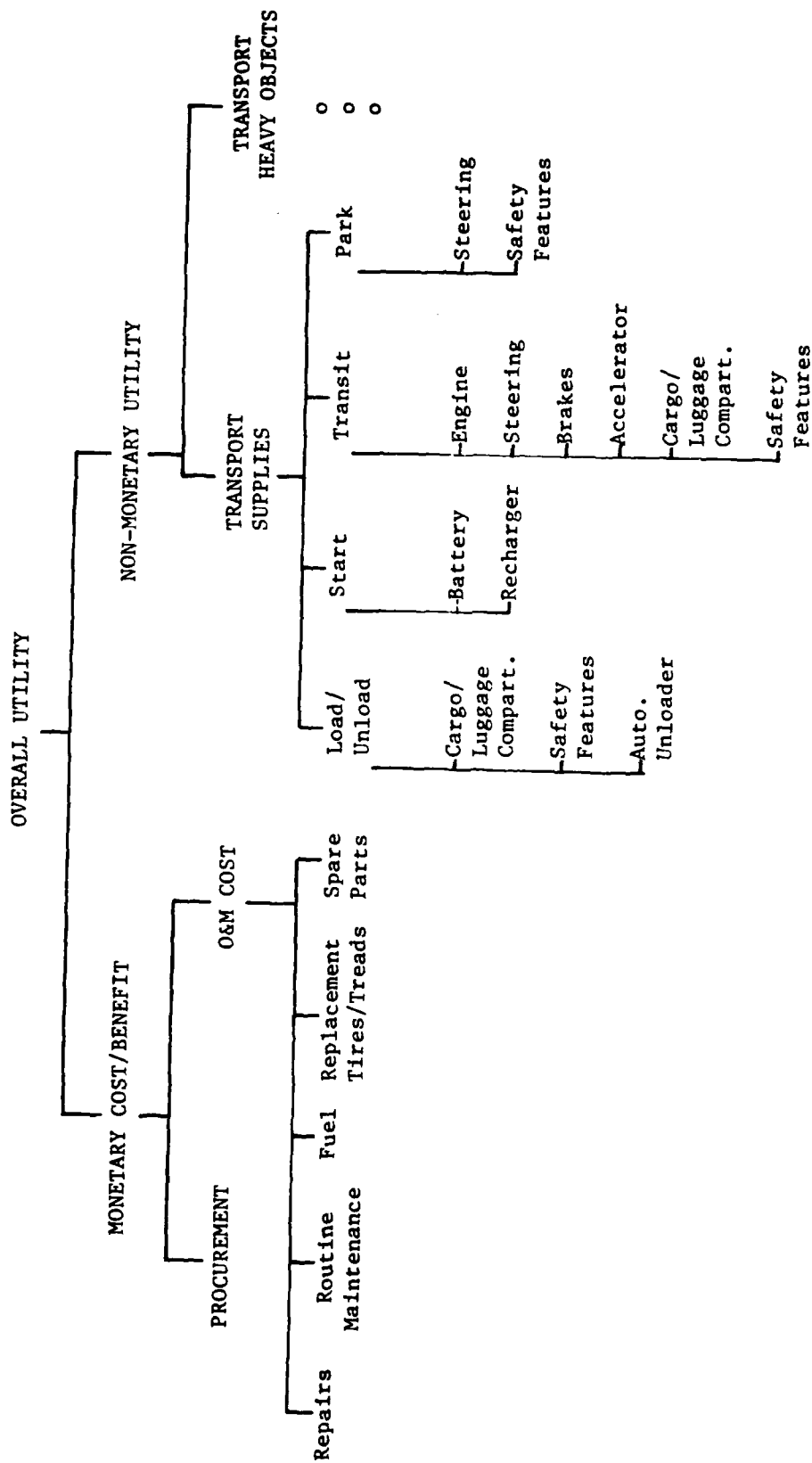


Figure 6-1  
HIERARCHICAL UTILITY STRUCTURE FOR SAMPLE PROBLEM

#### 6.4 GenTree Summary

GenTree is at present a concept of how machines can aid in decision analysis, rather than a specific procedure for implementing the concept. If any benefit is to be gained by the transfer of learning from one substantive problem to another, some GenTree-like computer program will be of great value in achieving a near-optimal ratio between the value of information transferred and the costs (time, manpower, effort, computer usage, etc.) of data-base manipulations. This optimum represents a compromise between two less satisfactory extremes: on one hand, a universally applicable system which can capture and use very little specific substantive information; and on the other, a system which can remember highly specific details about past analyses, but can manipulate, generalize, and apply such details only with great difficulty.

As the discussion in this chapter has indicated, GenTree as currently conceived could utilize more economical concepts that derive from prior experience in the fields of human learning and memory, artificial intelligence, and decision-analytic practice. The net result of a GenTree-like model would be a highly structured data management system which could incorporate substantive information into a formal representation framework, for subsequent recovery and application to related problems.

In the future, as some experience is gained in attempting to implement a GenTree-like program, and as more of the currently developing peripherally related technologies (e.g., artificial intelligence, interactive graphics, video disk storage) are assimilated, it should become possible to provide the user with a major head start at analyzing new problems. Furthermore, the continuing interaction between



GenTree technology and the peripherally related technologies should add to the capabilities of GenTree itself, and stimulate the growth of those other fields as well.

## 7.0 CONCLUSION

This report has presented two different concepts for aiding the decision maker without specialized decision-analytic training. QVAL, which has been implemented on an IBM 5110 computer, is the simpler approach, and represents an effort to stimulate the role of the trained decision analyst, while making extensive use of the decision maker's substantive expertise; GenTree, on the other hand, is a more complex system (much further from realization as an actual program), which augments the role of the content-neutral decision analyst by maintaining and utilizing an extensive substantive data base, whose contents may help to steer the analysis in an efficient direction.

QVAL's role represents a major step beyond the simple clerical and computational tasks associated with earlier decision aids. Guided by decision-analytic procedures and their information requirements, QVAL assumes an active role in probing the user's memory to ensure (so far as possible) elicitation of a sufficiently complete set of attributes, together with the scores and weights needed for a consistent evaluation of a set of options. Because it is constrained by definition to apply equally to all substantive areas, QVAL depends primarily on mnemonic procedures and on the user's own breadth of knowledge for its success.

GenTree is a concept which would greatly extend the role of a decision aid, even beyond what QVAL attempts, by participating actively in the decision-structuring process, not only as a neutral guide-clerk-calculator, but also as a source of information about the substantive problem and any related problems that have been encountered in the past. Data from prior learning and experience can be encoded so as

to be applicable in a variety of related instances, and retrieved by reference to a semantic organization based on the kinds of association typically encountered when one problem is recognized as somehow similar or related to a previously encountered one. A successful implementation of a GenTree program, together with a suitable data base, would constitute a major asset to any decision maker, both as a decision aid and as a source of information.

In the immediate future, it is possible to identify a few directions in which QVAL and GenTree might profitably develop. In the case of QVAL, probably the most important goal should be to improve interaction with the untrained user through a continuing process of refinement, field testing, and further refinement, until an untrained user can simply start the program and then follow instructions without the need to refer to an operating manual or a technical description. In order to reach this level of interactive flexibility (without forcing the experienced user through a detailed, repetitious routine), QVAL must develop a multi-level capability whereby the user can control the trade-off between speed of operation and completeness of descriptive and tutorial material. Again, much of this sort of development can be achieved only through performance studies, continued fine-tuning, and extensive experience.

GenTree, which has not been realized at all on an actual computer, could profit from further conceptual development, including reactions to any response generated by the description in Section 6.0. Once the details of its operation can be fully specified, and the routines for interacting with users and for updating the data base can be determined, a full-scale programming effort could be performed which would attempt to realize the GenTree system on an interactive computer. Input from members of the artificial intelligence, computer graphics, and other related

communities would be of particular value at this stage of development, in order to make the best use of currently available technology (and to identify promising areas for future developments). Naturally, once a GenTree system is implemented, it will need to be thoroughly tested, and refined or modified until it best meets the needs of its users.

Although they can be evaluated for their own sakes, QVAL and GenTree can probably best be regarded as first steps in a more extended process of development, testing, and feedback, iterated to achieve successively better approximations to the elusive goal of the ideal computerized decision aid. As related technologies for interaction, display, and memory organization develop, and as their current findings are assimilated, progress in decision aiding may take advantage of the new techniques, while at the same time providing insights and experience which can identify new needs and possibilities.

## REFERENCES

- Adams, J. L. Conceptual Blockbusting. San Francisco: W. H. Freeman and Company, 1974.
- Allardyce, L. B.; Amey, D. M.; Feuerwerger, P. H.; and Gulick, R. M. Documentation of Decision Aiding Software: EVAL Users Manual. McLean, VA: Decisions and Designs, Incorporated, November 1979.
- Barclay, S. and Randall, L. S. Satellite Surveillance Avoidance Optimization Aid. Final Report PR 77-14-36. McLean, VA: Decisions and Designs, Incorporated, December 1977.
- Collins, A. and Quillian, M. R. "Retrieval Time from Semantic Memory." Journal of Verbal Learning and Verbal Behavior, Vol. 8, 240-247, 1969.
- Fillmore, C. "The Case for Case." In Bach and Harms (eds.), Universals in Linguistic Theory. New York: Holt, Rinehart, and Winston, 1968.
- Gulick, R. M.; Kuskey, K. P.; and Buede, D. M. An Analysis of Investment Opportunities for Night and In-Weather Tactical Air Operations. Final Technical Report PR 79-25-309. McLean, VA: Decisions and Designs, Incorporated, December 1979.
- Hartigan, J. A. Clustering Algorithms. New York: Wiley, 1975.
- Humphreys, P. C. and Wisudha, A. MAUD - An Interactive Computer Program for the Structuring, Decomposition, and Recomposition of Preference between Multiattributed Alternatives. Decision Analysis Unit Technical Report 79-2, Uxbridge, Middlesex, UK: Brunel Institute of Organization and Social Studies, Brunel University, 1979.
- Kelly, G. A. A Theory of Personality. New York: Norton, 1963.
- Miller, G. A. "The Magical Number Seven Plus or Minus Two." Psychological Review, Vol. 63, 81-97, 1956.
- Negroponte, N. Spatial Data-Management. Cambridge, MA: Massachusetts Institute of Technology, 1979.
- Schank, R. "Conceptual Dependency: A Theory of Natural Language Understanding." Cognitive Psychology, Vol. 3, No. 4, 1972.

- Schank, R. "Identification of Conceptualizations Underlying Natural Language." Chapter in Schank, R. and Colby, K. M. (eds.), Computer Models of Thought and Language. San Francisco: W. H. Freeman and Company, 1973.
- Simon, H. A. and Newell, A. Human Problem Solving. Englewood Cliffs, NJ: Prentice Hall, 1971.
- Sticha, P. J.; Weiss, J. J.; and Donnell, M. L. Evaluation and Integration of Imprecise Information. Final Report PR 79-21-90, McLean, VA: Decisions and Designs, Incorporated, October 1979.
- Von Neumann, J. and Morgenstern, O. Theory of Games and Economic Behavior. Princeton, NJ: Princeton University Press, 1953.
- Von Winterfeldt, D. Structuring Decision Problems. Invited paper presented at the Seventh Research Conference on Subjective Probability, Utility, and Human Decision Making. Goteborg, Sweden, August 1979.
- Watson, S. R.; Weiss, J. J.; and Donnell, M. L. "Fuzzy Decision Analysis." IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, 1-9, 1979.
- Weizenbaum, J. "ELIZA--A Computer Program for the Study of Natural Language Communication between Man and Machine." Communications of the ACM, Vol. 9, No. 1, 36-45, January 1965.
- Winograd, T. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. Unpublished doctoral dissertation, Department of Mathematics, Massachusetts Institute of Technology, 1970.
- Winograd, T. "A Program for Understanding Natural Language." Cognitive Psychology, Vol. 3, 1972.
- Zadeh, L. A. "Fuzzy Sets." Information and Control, Vol. 8, 338-353, 1965.

APPENDIX A  
IBM 5110 EQUIPMENT AND OPERATION

A.1 Equipment

The computer used for implementing QVAL is the IBM 5110 portable computer (shown in Figure A-1). The following section briefly describes the equipment that you will use. Remember that there are no typewriter keys or switches that you can press that will damage the equipment. If you make a mistake, or an omission, there will always be the opportunity for corrections.

The IBM 5110 has five components which comprise the basic self-contained system illustrated in Figure A-1. These are the operator selection switches, display screen, keyboard, tape drive, and the central processor with its associated memory. In addition, a printer accompanies this basic unit.

A.1.1 Operator selection switches -

- o L32 64 R32 - This three-position switch allows the user to display the left 32 characters of the display (position L32), the right 32 characters (R32) or the entire display of 64 characters (64). Operation of the QVAL program requires that this switch be in the center position (64).
- o Reverse Display - Some users prefer viewing a black on white image to a white on black. The Reverse Display switch allows the user to select the type of image he prefers. It should be noted that reversal of the display will require a brightness adjustment. Also, the Reverse Display switch will not affect the image displayed on an auxiliary TV monitor.

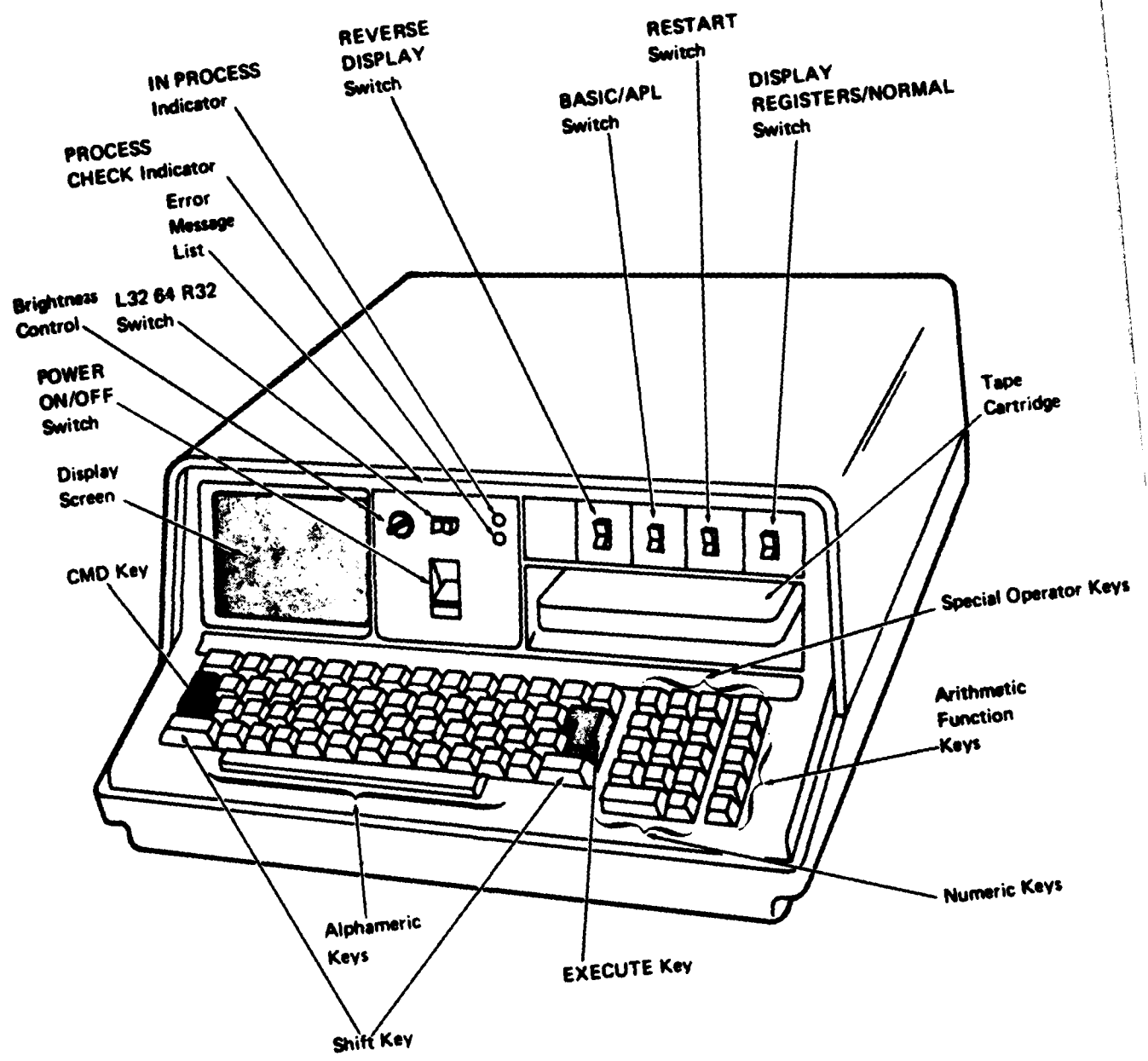


Figure A-1



- o BASIC/APL - The QVAL program is written in APL (A Programming Language) and this switch must be in the APL position. By placing the switch in the BASIC position, the computer is configured to operate in the BASIC language.
- o RESTART - The Restart switch is used to re-initialize the IBM 5110. Depressing this switch is equivalent to turning off power to the machine and restarting.
- o Display Registers/NORMAL - This switch should be in the NORMAL position when operating the QVAL program. The Display Registers position provides a display of internal machine code used in diagnostic testing of the machine.

A.1.2 The display - The display is a cathode ray tube (CRT) which allows 16 lines of data to be displayed. Each line may contain up to 64 characters. The computer scrolls each line from bottom to top. Lines that scroll off the top are lost. The display screen has two functions:

- a) As you type characters, these will appear on the bottom two lines of the screen. A flashing cursor (-) will indicate where the next character will be entered.
- b) The computer will help you organize and summarize the data that you enter. Tables of these data will be displayed on the upper 14 lines of the display.

When the 5110 is making computations, the screen will often go blank and the red (IN PROCESS) light will be illuminated.

A.1.3 The keyboard - The layout of the IBM 5110 keyboard is similar to that of a standard typewriter. As you will note, many of the keys have special symbols embossed over the standard typewriter characters. These symbols are used to write programs in the APL language and are not necessary when operating the QVAL program.

In addition to the standard keyboard, note that the 5110 has a numeric keypad, similar to that of an adding machine. These keys are interchangeable with the numbers appearing in the top row of the keyboard, and many users find them more convenient to use.

As characters are typed, they appear on the display at the location identified by the cursor. In general, this will occur on the bottom line of the display screen.

Finally, there are a number of additional keys that perform special functions. These keys are discussed below.

#### SHIFT

The SHIFT key performs the same function as a SHIFT key on a typewriter.

#### FORWARD SPACE

When this key is pressed once, the cursor moves one position to the right. When this key is held down, the cursor continues to move to the right. When the cursor reaches the last position on one input line, it goes to the first position on the next input line.

## BACKSPACE

When this key is pressed once, the cursor moves one position to the left. When it is held down, the cursor continues to move to the left. When the cursor reaches position 1 on one input line, it goes to the last position on the previous input line.

## HOLD

When pressed once, HOLD causes all processing to stop; when pressed again, it allows processing to resume. The primary purpose of HOLD is to permit reading the display information during an output operation, when the display is changing rapidly. When the hold is in effect (HOLD pressed once), only the COPY DISPLAY key is active.

## EXECUTE

When this key is pressed, the input line of information on the display screen is processed by the system. This key must be pressed for any input to be processed.

## ATTN

The ATTN key erases from both the computer's memory and the display screen everything beyond the space where the flashing cursor is positioned. It does not erase anything before the flashing cursor.

## INSERT

When the CMD key is held down and the FORWARD SPACE is pressed once, the characters at and to the

right of the cursor position (flashing character) are moved to the right one position, and a blank character is inserted at the cursor position. The cursor does not move. For example:

Before the insert operation: 123567

↓ Flashing character

After the insert operation: 123\_567

When these keys are both held down, the characters continue to move to the right and blank characters continue to be inserted.

#### DELETE

When the CMD key is held down and the backspace key is pressed once, the character at the cursor position (flashing character) is deleted and all characters to the right are moved over one position to the left to close up the space. The cursor is not moved. For example:

Before the delete operation: 1234456

↓ Flashing character

After the delete operation: 123456

When these keys are both held down, the characters at the cursor position continue to be deleted and all the characters to the right are moved to the left.

A.1.4 The magnetic diskette - This device is used to store the programs and data that you will use. Before starting the QVAL program, the program must be loaded from a diskette. Also, after a user has created a new model via the QVAL program,

the information concerning this new model must be stored on tape if it is to be used again. The QVAL program automatically handles the control of the disk drive, but the user must physically load the disk in the machine (Figure A-2).

A.1.5 The central processor and memory - The central processor is a microprocessor developed by IBM. This unit executes the commands stored in the computer's memory.

A.1.6 The printer - The printer, when turned on, will print a hard copy of the CRT display unless the program specifically directs it not to do so. In the case of QVAL, once the initial heading has been displayed, you will be instructed to turn the printer on (and then hit EXECUTE). From this point on, you should leave the printer on, as the QVAL program will control printout for you.

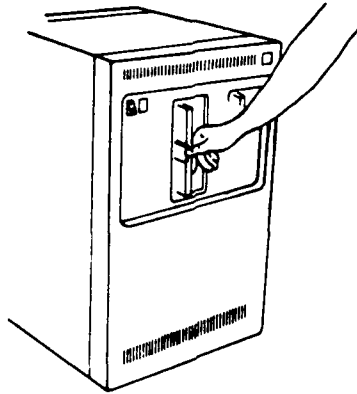
If at any time the information on the display screen seems important enough to record, a printed copy can be obtained by turning on the printer (if it is not already on), then simultaneously pressing the CMD button and the x (multiplication sign) button.

A.1.7 Potential problems - The QVAL program has many internal safeguards which should prevent most problems, but there are a few things that can go wrong.

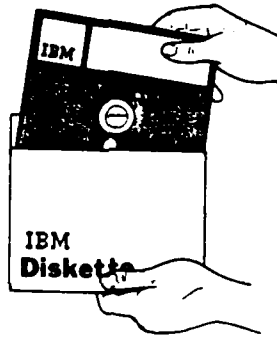
TYPING MISTAKE - The first problem that is likely to occur is that a user will mistype a response to the computer. This is very easy to correct. Prior to depressing the EXECUTE key, simply type over the portion of the response that is incorrect utilizing the SPACE FORWARD and SPACE BACK keys. Remember that the computer will not process a user response until the EXECUTE key is depressed.

To insert a diskette into the diskette drive:

1. Open the diskette drive cover.

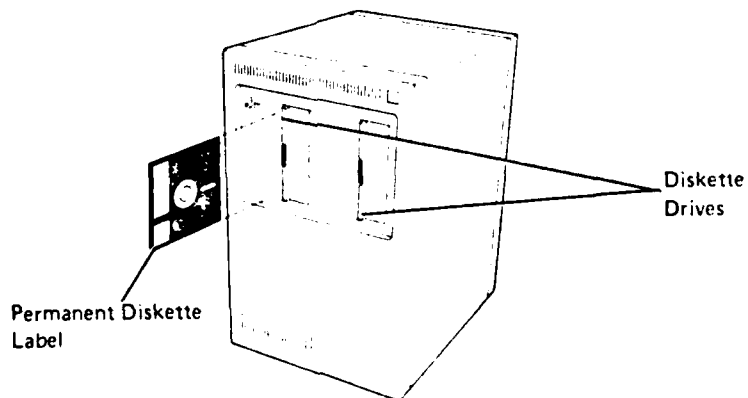


2. Remove the diskette from its envelope by grasping its upper edge.



3. Insert the diskette into the diskette drive.

Note: The permanent diskette label must be in the lower right corner as the diskette is inserted into the diskette drive.



4. Close the diskette drive cover only after the diskette is fully inserted.

Figure A-2

DISKETTE IMPROPERLY LOADED - If the magnetic disk is not properly loaded in the disk drive, QVAL will not execute properly. To correct this situation, simply remove the disk, make sure it is correctly positioned, reload the disk, depress the RESTART to clear the computer, and start over (see Figure A-2).

PROCESS CHECK ERROR - If the Process Check Light (Figure A-1) comes on, the computer has encountered internal problems. Depress RESTART and try again. If the light comes on again, an IBM service representative should be notified.

## A.2 Starting Up the System

Turn on the IBM 5110 and check the position of the operator selection switches. If the printer is to be used for recording the output displays, make sure that it is connected properly prior to turning on the 5110. Do not connect or disconnect the printer in the middle of any operation. To connect the printer, screw the box-like appendage of the printer into the back of the 5110. Be sure that the printer is plugged in.

When the computer has completed its internal check, the following display will appear in the lower left of the display screen:

CLEAR WS

Insert the QVAL program diskette and type the following instruction exactly as shown:

)LOAD QVAL

Press the EXECUTE key and the computer will load the QVAL program from the diskette. When this process is completed, the computer will display the following headings:

```
*****
*
*          QVAL - QUICK EVALUATION ROUTINE
*
*
*****
```

TURN ON PRINTER. PRESS EXECUTE TO CONTINUE...

At this point, the QVAL system is loaded and ready to begin.

### A.3 Bringing Down the System

When the user is finished with his work, he can bring down the system by removing his diskette, turning the computer power off, and turning the printer off. This can be done whenever the computer is waiting for a user input, but should not be done while the printer is operating or while the disk drive is storing or retrieving a model.



**CONTRACT DISTRIBUTION LIST**  
**(Unclassified Technical Reports)**

Director Advanced Research Projects Agency Attention: Program Management Office 1400 Wilson Boulevard Arlington, Virginia 22209	2 copies
Office of Naval Research Attention: Code 455 800 North Quincy Street Arlington, Virginia 22217	3 copies
Defense Technical Information Center Attention: DDC-TC Cameron Station Alexandria, Virginia 22314	12 copies
DCASMA Baltimore Office Attention: Mr. K. Gerasim 300 East Joppa Road Towson, Maryland 21204	1 copy
Director Naval Research Laboratory Attention: Code 2627 Washington, D.C. 20375	6 copies
Office of Naval Research Boston Branch Office 666 Summer Street Boston, Massachusetts 02210	1 copy

APRIL 1980

**SUPPLEMENTAL DISTRIBUTION LIST**  
**(Unclassified Technical Reports)**

Department of Defense

Assistant Director (Net Technical Assessment)  
Office of the Deputy Director of Defense  
Research and Engineering (Test and  
Evaluation)

The Pentagon, Room 3C125  
Washington, DC 20301

Director, Defense Advanced Research  
Projects Agency  
1400 Wilson Boulevard  
Arlington, VA 22209

Director, Cybernetics Technology Office  
Defense Advanced Research Projects Agency  
1400 Wilson Boulevard  
Arlington, VA 22209

Director, ARPA Regional Office (Europe)  
Headquarters, U.S. European Command  
APO New York 09128

Director, ARPA Regional Office (Pacific)  
Staff CINCPAC, Box 13  
Camp H. M. Smith, Hawaii 96861

Dr. Don Hirta  
Defense Systems Management School  
Building 202  
Ft. Belvoir, VA 22060

Chairman, Department of Curriculum  
Development  
National War College  
Ft. McNair, 4th and P Streets, SW  
Washington, DC 20319

Defense Intelligence School  
Attention: Professor Douglas E. Hunter  
Washington, DC 20374

Vice Director for Production  
Management Office (Special Actions)  
Defense Intelligence Agency  
Room 1E863, The Pentagon  
Washington, DC 20301

Command and Control Technical Center  
Defense Communications Agency  
Attention: Mr. John D. Hwang  
Washington, DC 20301

Department of the Navy

Office of the Chief of Naval Operations  
(OP-951)  
Washington, DC 20450

Office of Naval Research  
Assistant Chief for Technology (Code 200)  
800 N. Quincy Street  
Arlington, VA 22217

Office of Naval Research (Code 230)  
800 North Quincy Street  
Arlington, VA 22217

Office of Naval Research  
Naval Analysis Programs (Code 431)  
800 North Quincy Street  
Arlington, VA 22217

Office of Naval Research  
Operations Research Programs (Code 434)  
800 North Quincy Street  
Arlington, VA 22217

Office of Naval Research  
Information Systems Program (Code 437)  
800 North Quincy Street  
Arlington, VA 22217

Director, ONR Branch Office  
Attention: Dr. Charles Davis  
536 South Clark Street  
Chicago, IL 60605

Director, ONR Branch Office  
Attention: Dr. J. Lester  
495 Summer Street  
Boston, MA 02210

Director, ONR Branch Office  
Attention: Dr. E. Gloye  
1030 East Green Street  
Pasadena, CA 91106

Director, ONR Branch Office  
Attention: Mr. R. Lawson  
1030 East Green Street  
Pasadena, CA 91106

Office of Naval Research  
Scientific Liaison Group  
Attention: Dr. M. Bertin  
American Embassy - Room A-407  
APO San Francisco 96503

Dr. A. L. Slafkosky  
Scientific Advisor  
Commandant of the Marine Corps (Code RD-1)  
Washington, DC 20380

Headquarters, Naval Material Command  
(Code 0331)  
Attention: Dr. Heber G. Moore  
Washington, DC 20360

Dean of Research Administration  
Naval Postgraduate School  
Attention: Patrick C. Parker  
Monterey, CA 93940

Superintendent  
Naval Postgraduate School  
Attention: R. J. Roland, (Code 52R1)  
C<sup>3</sup> Curriculum  
Monterey, CA 93940

Naval Personnel Research and Development  
Center (Code 305)  
Attention: LCDR O'Bar  
San Diego, CA 92152

Navy Personnel Research and Development  
Center  
Manned Systems Design (Code 311)  
Attention: Dr. Fred Muckler  
San Diego, CA 92152

Naval Training Equipment Center  
Human Factors Department (Code N215)  
Orlando, FL 32813

Naval Training Equipment Center  
Training Analysis and Evaluation Group  
(Code N-00T)  
Attention: Dr. Alfred F. Smode  
Orlando, FL 32813

Director, Center for Advanced Research  
Naval War College  
Attention: Professor C. Lewis  
Newport, RI 02840

Naval Research Laboratory  
Communications Sciences Division (Code 5403)  
Attention: Dr. John Shore  
Washington, DC 20375

Dean of the Academic Departments  
U.S. Naval Academy  
Annapolis, MD 21402

Dr. Glen R. Algaier  
Head  
Command & Control  
Information Processing Branch  
Naval Ocean Systems Center  
San Diego, CA 92152

Chief, Intelligence Division  
Marine Corps Development Center  
Quantico, VA 22134

#### Department of the Army

Deputy Under Secretary of the Army  
(Operations Research)  
The Pentagon, Room 2E621  
Washington, DC 20310

Director, Army Library  
Army Studies (ASDIRS)  
The Pentagon, Room 1A534  
Washington, DC 20310

U.S. Army Research Institute  
Organizations and Systems Research Laboratory  
Attention: Dr. Edgar M. Johnson  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Director, Organizations and Systems  
Research Laboratory  
U.S. Army Institute for the Behavioral  
and Social Sciences  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Technical Director, U.S. Army Concepts  
Analysis Agency  
8120 Woodmont Avenue  
Bethesda, MD 20014

Director, Strategic Studies Institute  
U.S. Army Combat Developments Command  
Carlisle Barracks, PA 17013

Commandant, Army Logistics Management Center  
Attention: DRXMC-LS-SCAD (ORSA)  
Ft. Lee, VA 23801

Department of Engineering  
United States Military Academy  
Attention: COL A. F. Grum  
West Point, NY 10996

Marine Corps Representative  
U.S. Army War College  
Carlisle Barracks, PA 17013

Chief, Studies and Analysis Office  
Headquarters, Army Training and Doctrine  
Command  
Ft. Monroe, VA 23351

Commander, U.S. Army Research Office  
(Durham)  
Box CM, Duke Station  
Durham, NC 27706

Department of the Air Force

Assistant for Requirements Development  
and Acquisition Programs  
Office of the Deputy Chief of Staff for  
Research and Development  
The Pentagon, Room 4C331  
Washington, DC 20330

Air Force Office of Scientific Research  
Life Sciences Directorate  
Building 410, Bolling AFB  
Washington, DC 20332

Commandant, Air University  
Maxwell AFB, AL 36112

Chief, Systems Effectiveness Branch  
Human Engineering Division  
Attention: Dr. Donald A. Topmiller  
Wright-Patterson AFB, OH 45433

Deputy Chief of Staff, Plans, and  
Operations  
Directorate of Concepts (AR/XOCCC)  
Attention: Major R. Linhard  
The Pentagon, Room 4D 1047  
Washington, DC 20330

Director, Advanced Systems Division  
(AFHRL/AS)

Attention: Dr. Gordon Eckstrand  
Wright-Patterson AFB, OH 45433

Commander, Rome Air Development Center  
Attention: Mr. John Atkinson  
Griffis AFB  
Rome, NY 13440

IRD, Rome Air Development Center  
Attention: Mr. Frederic A. Dion  
Griffis AFB  
Rome, NY 13440

HQS Tactical Air Command  
Attention: LTCOL David Dianich  
Langley AFB, VA 23665

Colonel Margenthaler  
AFIT/EN  
Wright-Patterson AFB, OH 45433

LTCOL Robert G. Gough  
AFTEC/DAY  
Curtland AFB, NM 87117

Other Government Agencies

Chief, Strategic Evaluation Center  
Central Intelligence Agency  
Headquarters, Room 2G24  
Washington, DC 20505

Director, Center for the Study of  
Intelligence  
Central Intelligence Agency  
Attention: Mr. Dean Moor  
Washington, DC 20505

Mr. Richard Heuer  
27585 VIA Sereno  
Carmel, CA 93923

Office of Life Sciences  
Headquarters, National Aeronautics and  
Space Administration  
Attention: Dr. Stanley Deutsch  
600 Independence Avenue  
Washington, DC 20546

Other Institutions

Department of Psychology  
The Johns Hopkins University  
Attention: Dr. Alphonse Chapanis  
Charles and 34th Streets  
Baltimore, MD 21218

Institute for Defense Analyses  
Attention: Dr. Jesse Orlansky  
400 Army Navy Drive  
Arlington, VA 22202

Director, Social Science Research Institute  
University of Southern California  
Attention: Dr. Ward Edwards  
Los Angeles, CA 90007

Perceptronics, Incorporated  
Attention: Dr. Amos Freedy  
6271 Variel Avenue  
Woodland Hills, CA 91364

Stanford University  
Attention: Dr. R. A. Howard  
Stanford, CA 94305

Director, Applied Psychology Unit  
Medical Research Council  
Attention: Dr. A. D. Baddeley  
15 Chaucer Road  
Cambridge, CB 2EF  
England

Department of Psychology  
Brunel University  
Attention: Dr. Lawrence D. Phillips  
Uxbridge, Middlesex UB8 3PH  
England

Decision Analysis Group  
Stanford Research Institute  
Attention: Dr. Miley W. Markhofer  
Menlo Park, CA 94025

Decision Research  
1201 Oak Street  
Eugene, OR 97401

Department of Psychology  
University of Washington  
Attention: Dr. Lee Roy Beach  
Seattle, WA 98195

Department of Electrical and Computer  
Engineering  
University of Michigan  
Attention: Professor Kan Chen  
Ann Arbor, MI 94135

Department of Government and Politics  
University of Maryland  
Attention: Dr. Davis B. Bobrow  
College Park, MD 20747

Department of Psychology  
Hebrew University  
Attention: Dr. Amos Tversky  
Jerusalem, Israel

Dr. Andrew P. Sage  
School of Engineering and Applied  
Science  
University of Virginia  
Charlottesville, VA 22901

Professor Raymond Tanter  
Political Science Department  
The University of Michigan  
Ann Arbor, MI 48109

Professor Howard Raiffa  
Morgan 302  
Harvard Business School  
Harvard University  
Cambridge, MA 02163

Department of Psychology  
University of Oklahoma  
Attention: Dr. Charles Gettys  
455 West Lindsey  
Dale Hall Tower  
Norman, OK 73069

Institute of Behavioral Science #3  
University of Colorado  
Attention: Dr. Kenneth Hammond  
Room 201  
Boulder, Colorado 80309